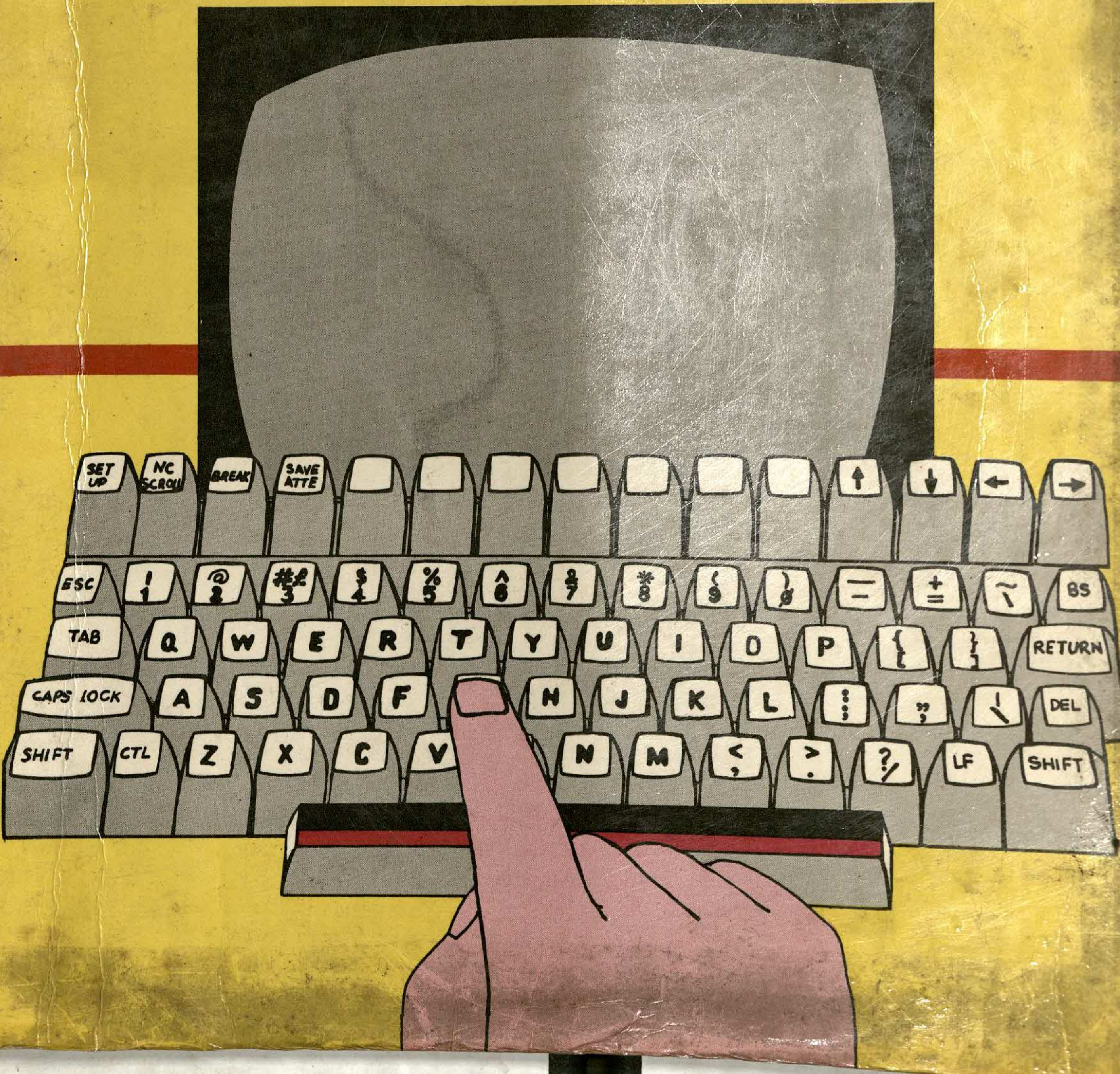
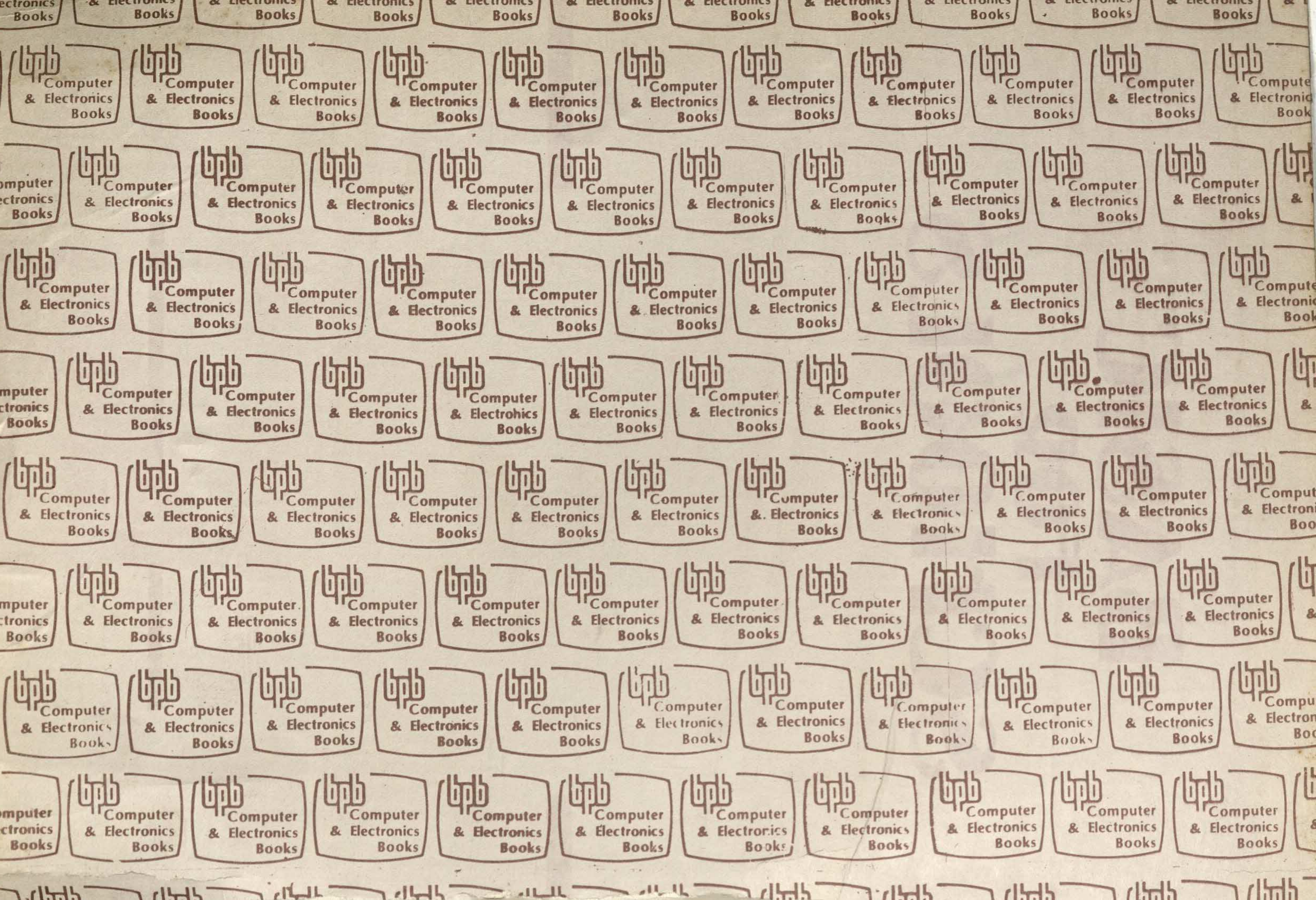
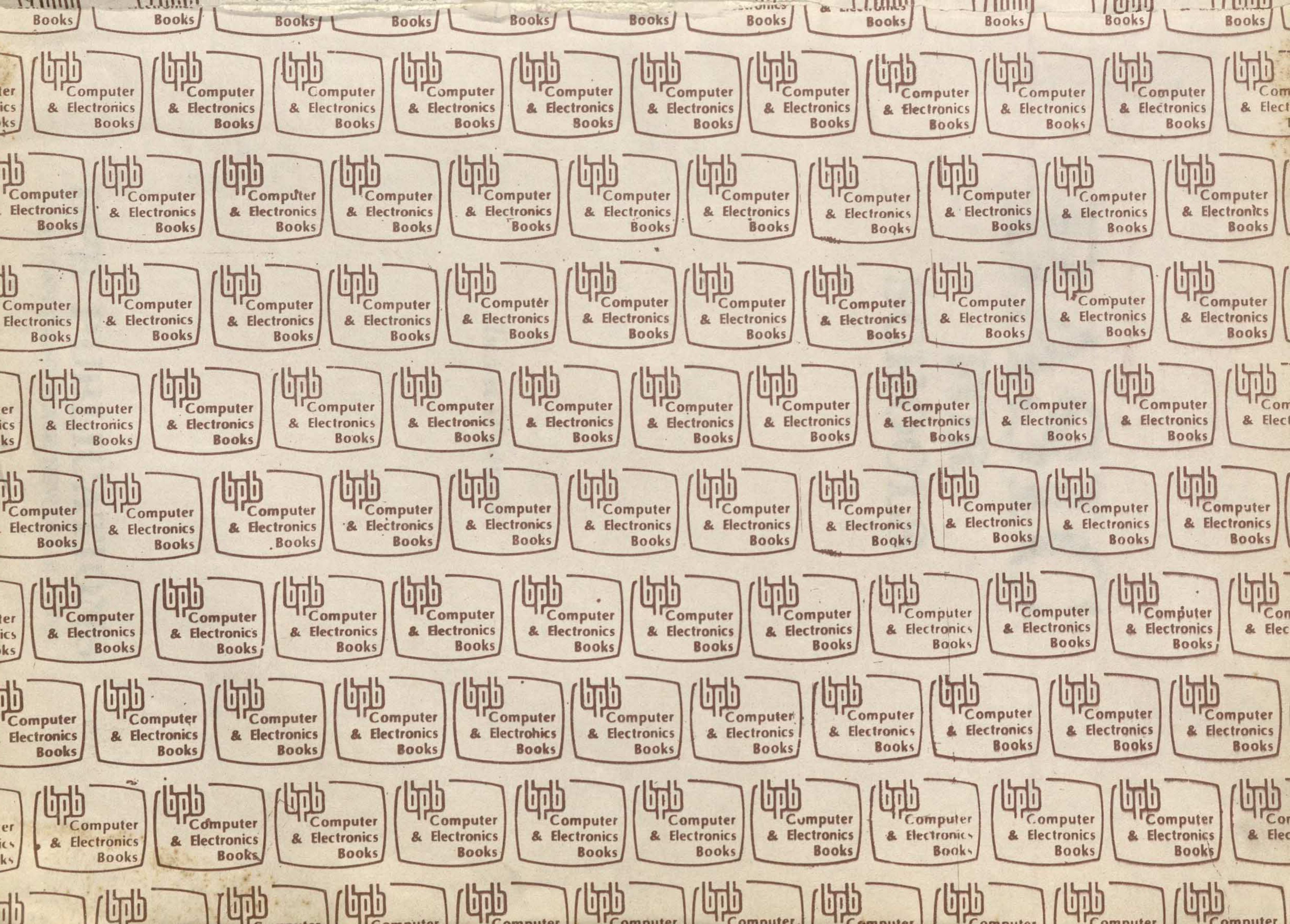
BASIC FOR SCHOOLS




bpb
Computer
& Electronics
Books


bpb
Computer
& Electronics
Books

✓

BASIC
FOR
SCHOOLS

Janet D'souza

BPB PUBLICATIONS
B-14, CONNAUGHT PLACE, NEW DELHI-110001.

© BPB PUBLICATIONS, New Delhi

FIRST EDITION–1991

Distributors:

BPB BOOK CENTRE
376, Old Lajpat Rai Market, Delhi–110006
BUSINESS PROMOTION BUREAU
8/1, Ritchie Street, Mount Road, Madras–600002
BUSINESS PROMOTION BUREAU
4-3-269, Giriraj Lane, Bank Street, Hyderabad–500001
COMPUTER BOOK CENTRE
12, Shrungar Complex, M. G. Road, Bangalore–560001

Acc No – 16843

Notice

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the written permission of BPB Publications.

Published by Manish Jain for BPB Publications, B-14, Connaught Place,
New Delhi – 110001 and Printed by him at Pressworks, Delhi – 110007.

To

My dear parents Jerdme and Lena Lobo who inspired me the most.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Br Pinto, Br Foley and Br Norohna for carefully going through my text. Br Gaffney and Br D'souza for rendering me their guidence.

My heartfelt thanks to Mewin (husband), Lancy (brother) and Maxim (brother in law) for helping me with the editing of the manuscript.

Last but not the least Mr. Manish Jain deserves thanks for publishing my book.

Janet D'souza,
Computer Teacher,
St. Columba's School,
New Delhi.

Dated: 15th May'91

FOREWORD

o one is really aware of how far-reaching the effects of one's actions an be!

When we pioneered computers in St. Columba's School, New Delhi, e little dreamed that our experiment would sweep faster than a rairie fire across the educational scene in the Country and over the ears the tempo has increased, the syllabus has been developed, new omponents added to it, more advanced technology introduced, and even he toddlers in the primary school have become adept at a manipulating achines with their little hands. To facilitate this, numerous books are eing written as authors rush into capitalise on a ready and flourishing arket. There is much pirated work much shoddy material. But in the idst of this literary jungle, some few books emerge born out of a eacher's actual experience. In these books lucid instructions cater to elt needs and make the subject not only simple but also enjoyable.

Mrs. Janet D'Souza's book is such a product.

J.P. Pinto
Provincial Superior
St. Columba's School
New Delhi.

ated: 2nd May'91

CONTENTS

		Page
1.	INTRODUCTION	1
2.	BASIC	17
3.	SIMPLE CALCULATIONS	29
4.	LET AND REM STATEMENT	41
5.	CORRECTIONS OF BUGS	53
6.	FLOWCHARTING	55
7.	INPUT STATEMENT	65
8.	GOTO STATEMENT	73
9.	FUNCTIONS	81
10.	REFORMING STRINGS	93
11.	IF–THEN STATEMENT	101
12.	FOR–NEXT STATEMENT	113
13.	FUN PROGRAMS	123
14.	GLOSSARY	129
15.	ANSWERS	133

1

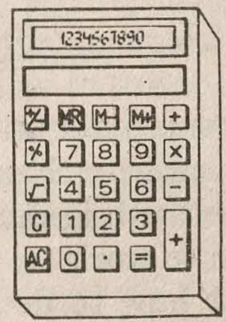
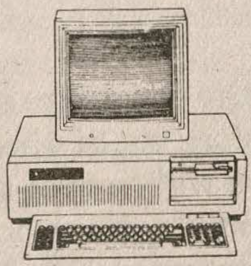
Introduction

This is the age of computers. The word "Computer" itself means that which

- computes
- solves
- processes

This device (machine) that processes information is called *computer*.

Man is the wonder of wonders and (no wonder he) is aiming at the best. Computer is the result of his ultimate achievements. Just as human beings are blessed with a brain, computers are a blessing from human efforts.

Calculator	Computer
	

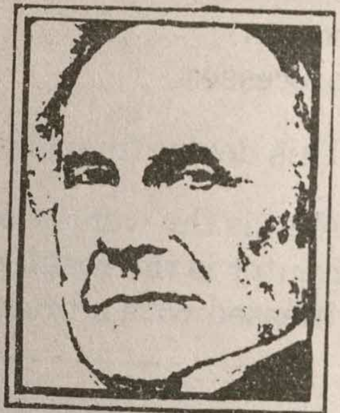
I am sure, you know a calculator. It adds, subtracts, divides and multiplies. It is a device specially designed to do only mathematical calculations. A computer is some-what like a calculator, but with more marvelous (great) features.

To a calculator, a TV screen and a typewriter like keyboard is attached. These three put together work automatically to give answers to the questions being typed. It becomes what we call a computer. Therefore a computer is called a processor/information processing machine.

Calculators	Computers
1. It can do simple calculations.	1. It can also perform very complex calculations.
2. It works at a fast speed.	2. It works at a much faster speed than a calculator.
3. It cannot process letters & certain symbols that is * @ etc. It can perform numeric calculations only.	3. It can perform non-numeric processing also.
4. It has a small memory power.	4. It has a vast memory power.
5. It can perform arithmetic calculations.	5. It not only performs arithmetic calculations but also performs logical calculations.

As you all have a father, similarly Charles Babbage is known as the *father* of Modern Computers. He was a british mathematician on whose ideas the modern day computers have been developed.

Computers have come not to replace man but to assist him in his efforts to accumulate more and more information, for the betterment of man.



Charles Babbage

Some of the people are of the opinion that a computer is a hi-tech machine, and is therefore out of their reach and much smarter than them. The study of computers will go over their heads. But the truth is that the computer is simply an empty, dumb box without a person to work on it. It is just like a little inexperienced kid. It has to be taught how to do things. The only difference is that once taught, it will perform the given job diligently and perfectly without making any mistake. You may think it sounds like a big bang thing, but the fact is its intelligence is no more than that of a dragonfly or even lesser than that of an ant.

The study of computers is very interesting and ever changing. Once you start you would like to go deeper and deeper to know more and more about it and the techniques to master it. It is fun to learn to master the super machine *Computer*. Once you have a hold on it you can discover its mysterious utilities in various aspects of life.

You must all be wondering that a computer must be a mysterious box of secrets, but let me tell you a top secret. A computer is nothing without a man, it can never think on its own. Everything is to be instructed or fed. Only then you can get the result.

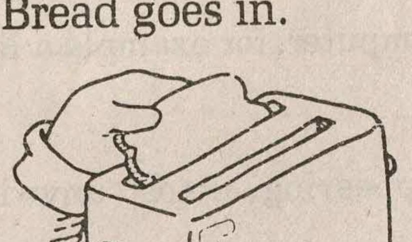
A computer does what it has been asked to do. It is possible to solve most difficult problem in a fraction of a second. A computer can solve any of your question provided you are able to program it, and is within the computer's limitation.

The main advantage of a Computer is the speed with which, it can do calculations. If you count, say 200 numbers in one minute, you would require 5000 minutes to count one million numbers, that is, 83 hours or in other words nearly 3½ days. A modern computer can do this in a few seconds.

In the years to come no one can afford to remain ignorant about computers and their working, so profound is going to be the impact of these machines on man.

Definition

Computer has opened up a road to all kinds of miracles. If you ask me, "What is a computer?" The answer would be that it is a box filled with electric wires, transistors, etc. etc. If you ask "How it works?". Well you do not need to know at this stage. Many people drive cars without knowing how they work. What they know is how to use them and how to drive them. This book will teach you how to "talk" to a computer in a way that both you and the computer understand, so that you can make the computer work for you; that is by means of a Program.



Bread goes in.

Toast comes out.

Speaking generally, it is an electronic device capable of accepting, storing the information in the form called *input,* processing (solving) it, and giving the results in the form called *output.*

In other words, the computer is a device that converts data into information. Data means facts. It is considered as the raw material of information. When data is properly arranged or organized it becomes Information.

INPUT	**PROCESS**	**OUTPUT**
Raw Vegetables	Cooking	Dish
Raw Materials	Processing	Finished Goods

Our brain is a super-computer. When a child joins a school, it never even knows even how to write. It completes its schooling and finally comes out as a literate person. In this process it has been given some information like General Knowledge, History, Geography, Social Science etc. that can be useful to him in his later life. This is stored in his memory for future use. In this way If input and process is good then output will obviously be good.

Therefore a machine which can accept input, manipulate and store data to output useful information in a desired form is a computer.

Works and Applications

A computer can be set to solve all kinds of mathematical problems. Also,

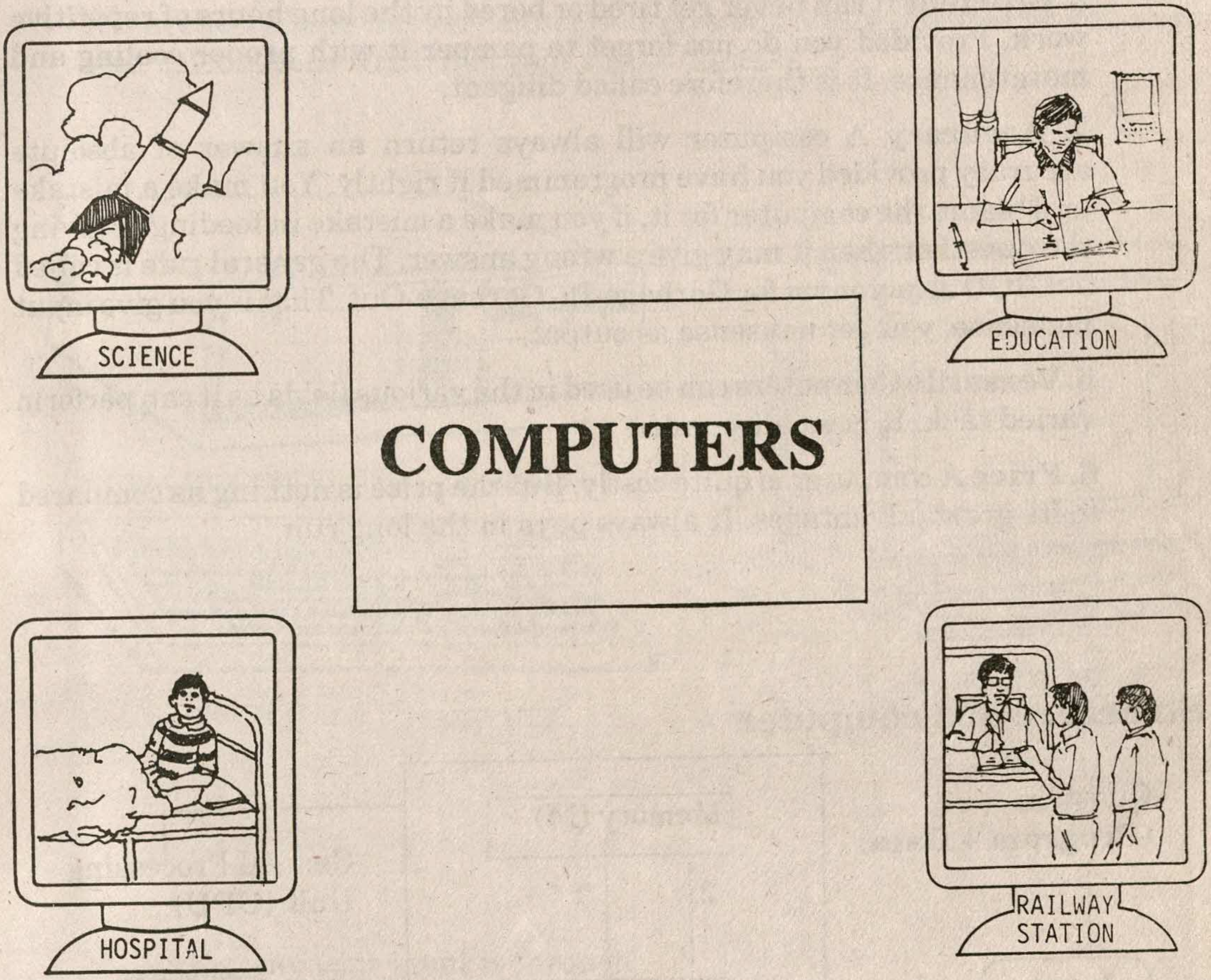
- It can be used to draw pictures and diagrams.
- It can distort or change the shape, size, colour etc. of a picture/diagram in any desired fashion.
- It can be used to compose music or generate exciting sound effects.
- It can 'talk' to one another on telephone with the help of a device called MOdulator DEModulator (MODEM)
- Machine could be controlled by a computer, for example a Robot.

Fields of Applications

Computers in the world of work are appearing almost anywhere and everywhere.

- Schools and Colleges
- Research Organisation
- Industries (Business)
- Hospitals

- Airport, Railway, Bus and other transport stations
- Government Departments
- Banks, and
- At homes too.



Characteristics/Features

1. Speed The foremost advantage of a computer is its great speed. You can never challenge the speed of a computer.

2. Storage It has an unlimited storage capacity for information.

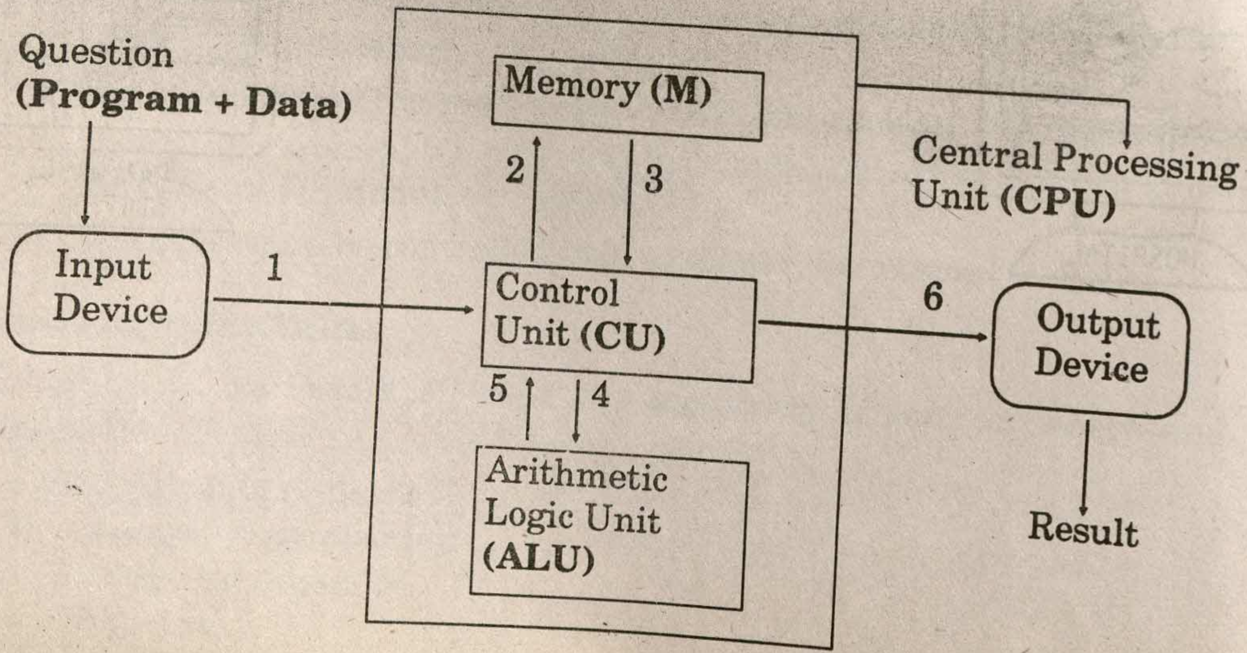
3. Efficient It can never get tired or bored by the long hours of repetitive work. Provided you do not forget to pamper it with proper cooling and maintenance. It is therefore called diligent.

4. Accuracy A computer will always return an answer of absolute accuracy provided you have programmed it rightly. You make a mistake and blame the computer for it, if you make a mistake in feeding or giving the question, then it may give a wrong answer. The general rule is called as GIGO an acronym for Garbage-In- Garbage-Out. That is you give input nonsense, you get nonsense as output.

5. Versatile Computers can be used in the various fields as it can perform varied task. It is an allrounder.

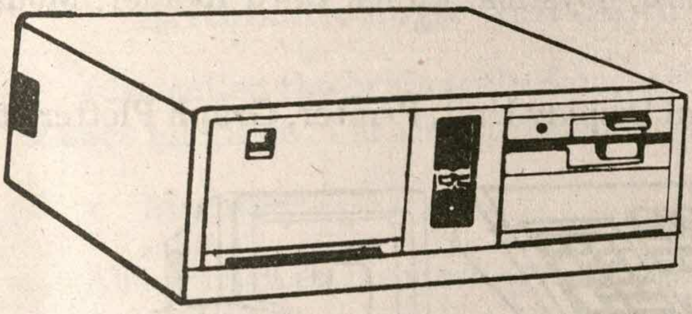
6. Price A computer is quite costly. But the price is nothing as compared to its great advantages. It always pays in the long run.

Structure of a computer

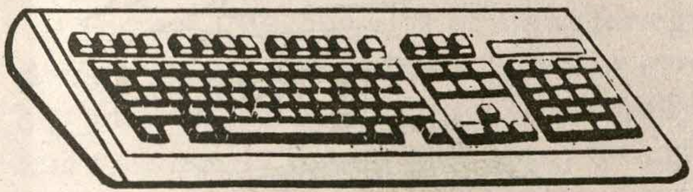


In this diagram the numbers and the arrows show the flow of data/information. And hence it is self-explanatory.

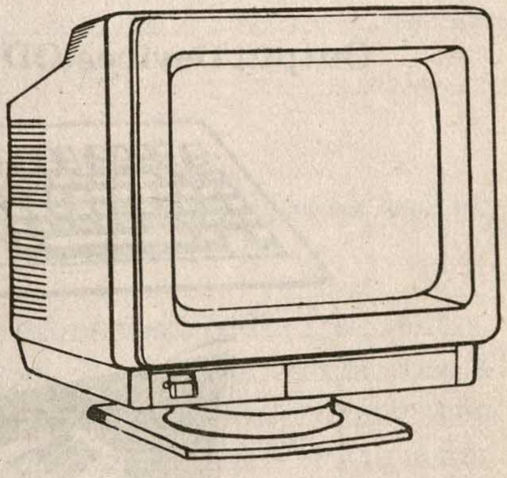
Whenever you are taught some new thing and later accept some questions/exercises, you gather it either by listening to it (ear) or by looking at the board (eyes). Later you solve the problem by using your brains. But when you are asked the answer, either you give out orally (mouth) or you show your calculations written in your copies (hands).



CENTRAL PROCESSING UNIT MAIN SYSTEM BOX



INPUT UNIT TYPEWRITER



OUTPUT UNIT
TV TYPE SCREEN

In this way your input is through

eyes and *ears*.

Processing unit is the brain and

output is through

hands or *mouth*.

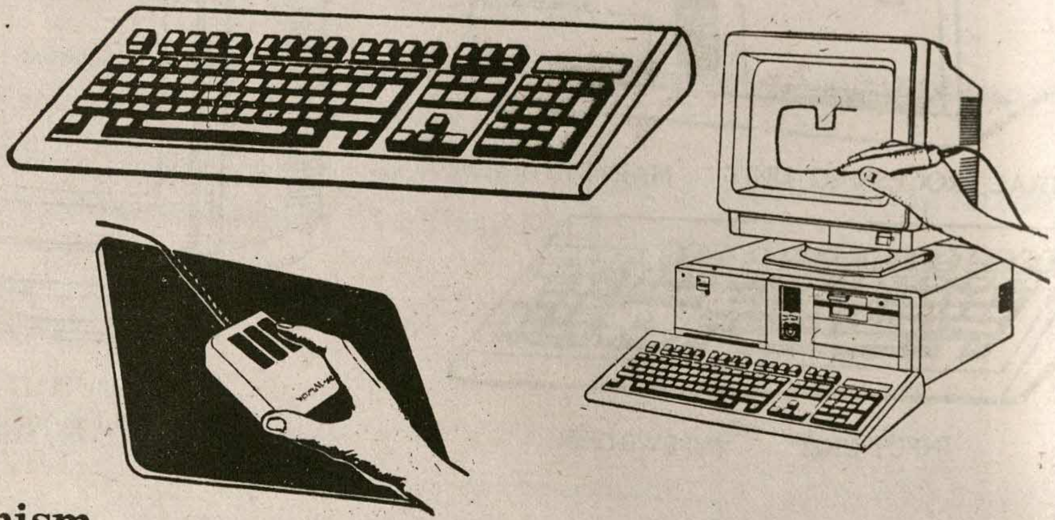
Similarly the nerve centre (brain) of any computer is its Central Processing Unit (CPU) a main solving part. But CPU cannot do anything on its own. It needs someone to feed in and receive the informations. The unit which accepts the instructions and some data is called the *input device.*

The unit which gives out the answer after calculations from the CPU is called *output device.*

There are different types of input and output devices used with a Computer. They are

Input Devices(ID) Keyboard, Joystick, Punch Card Reader, Mouse etc.

Output Devices(OD) Visual Display Unit, Printer, Graph Plotter etc.



Mechanism

When your teacher asks you any question, in a way she talks to you. That means communicating, or "talking". It is the most conversant medium to express one's view.

If you have a problem to solve by using a computer, you will have to 'talk' to it, that is, by typing out your problem on keyboard. That may not be enough as the computer will not follow your language. Therefore, you will have to type on the keyboard (ID) in a language the computer will understand.

It will very obediently accept any question you give but will not rebounce or throw back to you. After solving, it will give out the answer on the Monitor (VDU) (OD) provided the question is not beyond its limitations, and is properly worded in its language.

From the input device the data enters the CPU wherein it is sent to the memory. The memory stores the information and sends it for processing to ALU. The problem after being solved, the answer is again sent back to the memory. And whenever you instruct the computer to show the answer from the memory, the results are displayed through the output device (OD) the Monitor. But the whole transfer of data in the CPU is managed by the Control Unit.

Thus the **Central Processing Unit** is divided into 3 main parts:

- **Control Unit**
- **Arithmetic Logic Unit (ALU)**

Also called the brain (calculator) of the computer. As the name suggests it does all kinds of arithmetical and logical operations.

- **Memory**

Also known as the store room of the computer. As it stores raw as well as processed data.

You must be aware that many foreign delegations visit India frequently. Some of them do not know Hindi or even English. But this does not pose a big problem. As each of them is normally accompanied by an *Interpreter* who knows not only English but many foreign languages too. These VIPs in order to express their views or communicate, talk in their own language to the Interpreter and they in trun without changing the meaning talk to the Indian counterparts successfully. Similarly, when you type out your problem in any of the computer languages (also called as high level languages) namely BASIC, COBOL, PASCAL, FORTRAN etc. the computer will surely accept it but before processing, it will convert internally the above language into its own language called *machine language* or Binary (two numbers system.) Since a computer consists of circuits and components which can only be in two states That is when current is passing through them (called as 1) and when no current passes (called as 0). Therefore all alphabets, punctuations etc. are represented by a series of 1 and 0's.

Memory

It is also called the store room of the computer. It stores the instructions, that are to be processed, the data and the answer obtained after execution. There are 2 main types of memory namely:-

1. Primary
2. Secondary

- **Primary Memory** or internal memory is present within the CPU. It is the main memory of any computer. It is of 2 kinds : ROM & RAM.

  ROM Its full form is Read Only Memory. It's contents or instructions can only be read. It is required every time the computer is switched on. It is written or filled in by the manufacturer. It cannot be easily erased therefore, it is also called permanent memory.

  RAM Its full form is Random Access Memory. Its contents can be memory read as well as written anytime. Whatever is typed in by the user is also stored in the RAM. It can easily be crased. As soon as the computer is switched off, data stored in RAM is lost.

- **Secondary Memory** or External Memory is present outside the CPU. It is substitute or the additional memory of any computer.

  For example:- Magnetic disk, Winchester disk, Magnetic tapes etc.

Computer Terms

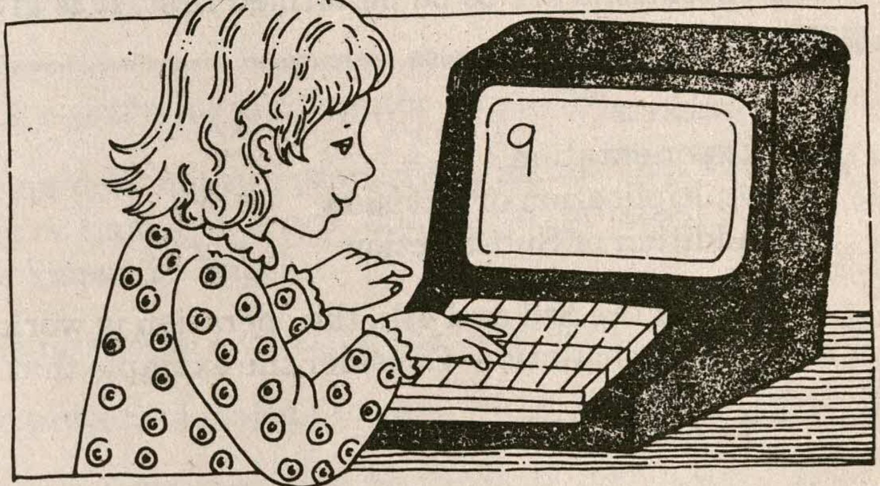
In English language a single unit is called a letter. Group of letters make a word, group of words in turn makes a sentence, group of related sentence is called a paragraph and many paragraphs make a passage etc.

Similarly, in Computers

- A single unit is called a character e.g. X, +, 3 etc.
- Group of characters is called a Field or an item.
- Group of items forms a Record.
- Group of records becomes a File.

Operators

Symbols or signs which help in performing computer operations are called "operators".



There are three types of operators.

Mathematical Operators Those dealing with arithmetical operations.

For example , -, +, /, *, ( ), ^

Relational Operators Those who show comparisons or relationship between two or more quantities (items).

For example >, <, >=, <=

Logical Operators Those which help in performing logical operations.

For example, AND, OR, NOT etc.

Most of the symbols used in computers are the same as used for mathematics, except for the following.

Symbols used in Operation	**General**	**Computer**
Multiplication	×	*
Division	÷	/
Exponentation	X^x	^ (e.g. 3^2)
Not equal to	≠	<>
Greater than or equal to	≥	>=
Less than or equal to	≤	<=

Remember, multiplication is never implied and division by zero is not allowed in computers. Unlike Bodmas (Brackets of division, multiplication, addition and subtraction) in Algebra, computer uses its own order to decide which operations are to be **performed** first. It is given in the order of their merit.

1. Brackets	( )
2. Exponentation	^
3. Multiplication or **Division**	* or /
4. Addition or Substr**action**	+ or -

In case of the 3rd and 4th, the operation is worked out from left to right through the calculation Consider the example that is given here:

```
7 * 4 / 2 + 5 - 2^3
```

First calculate 2 to the power of 3 = 8, then multiply 7 by 4 and divide the resultant by 2. Later add 5 to it and then subtract 8 from it. The answer is 11.

The same rule are followed even within the brackets. Let us see this example:

```
2 + (7 - 40/2^3 - 6 + 5)
```

We have solved them step by step for you to understand it better.

```
2 + (7 - 40/8 - 6 + 5)
2 + (7 - 5 - 6 + 5)
2 + (-4 + 5)
2 + (1)
3
```

Note: Use brackets wherever necessary to clarify any calculation in computers.

So far so good. Till now you must have got a better picture of a computer. We now present you with some brief definitions of terms used in computers. These would enable you a firm foundation on the subject before going further.

User The person who deals with or uses the computer.

Algorithm First step in solving any particular problem. Sequence of steps with all possible alternatives written in English language to solve a problem.

Programmer The person who writes a program in any Computer languag.

Computer Operator/User The person who operates or feeds the program into the computer.

Syntax To understand any language one has to know its grammer. Syntax is the word for *grammer* in computer languages.

Bugs If a program does not run properly, it means that there are some mistakes in writing or typing that are causing the program to misbehave. These mistakes are known as *Bugs*.

Debugging The process of correcting the bugs or errors in programs.

Executing The process of getting the answer or running a program.

Hardware What a computer is or whatever is inside the computer is Hardware. In other words, the physical parts of a computer system including the CPU, input and output devices come into the category of Hardware.

Liveware All the people who are concerned or related with (to) the functioning of computers are collectively called as Liveware. For example, Users, Programmers, etc.

Interface The connecting or the communication link between the CPU, the input and output device is an interface. It can be found in the form of wires, cables etc.

Software It is not possible to use a computer only with the hardware. What is required is the instruction. These instructions or programs are termed as Software.

Lastly, to judge whether or not you have understood this chapter in full, solve this exercise.

EXERCISE – 1

Q. I. Rearrange the following jumbled letters to form meaningful words by referring to the provided clues therein.

1. OURECTMP
(Machine that can accept, process or store information)
2. LACTOUROLA
(A dedicated computer to do only mathematical operations)
3. PITUN
(Information fed in the computer)
4. TAAD
(Raw facts and figures)
5. YMREOM
(Store room of the computer)
6. OERCDR
(Group of items)
7. PAROSROET
(Symbols used for calculations)
8. TRENIPR
(An example of output device)
9. ARGPROM
(A sequence of instructions for solving a given problem)
10. EUGINGDBG
(Process of eliminating the errors from a problem)

Q. II. Rewrite the following sentences by filling-in the blanks.

1........is the father of modern computers.
2........unit supervises the transfer of data within the CPU
3........is the symbol used for multiplication.
4........is the symbol used for not equal to.
5........devices are used to feed program in the computer.
6........is a computer term for a mistake.
7. The whole component system of a computer is called .....
8. The person who writes a program is called .....

Q. III. Give full forms of the following:

1. CPU
2. ALU

3. GIGO
4. CU

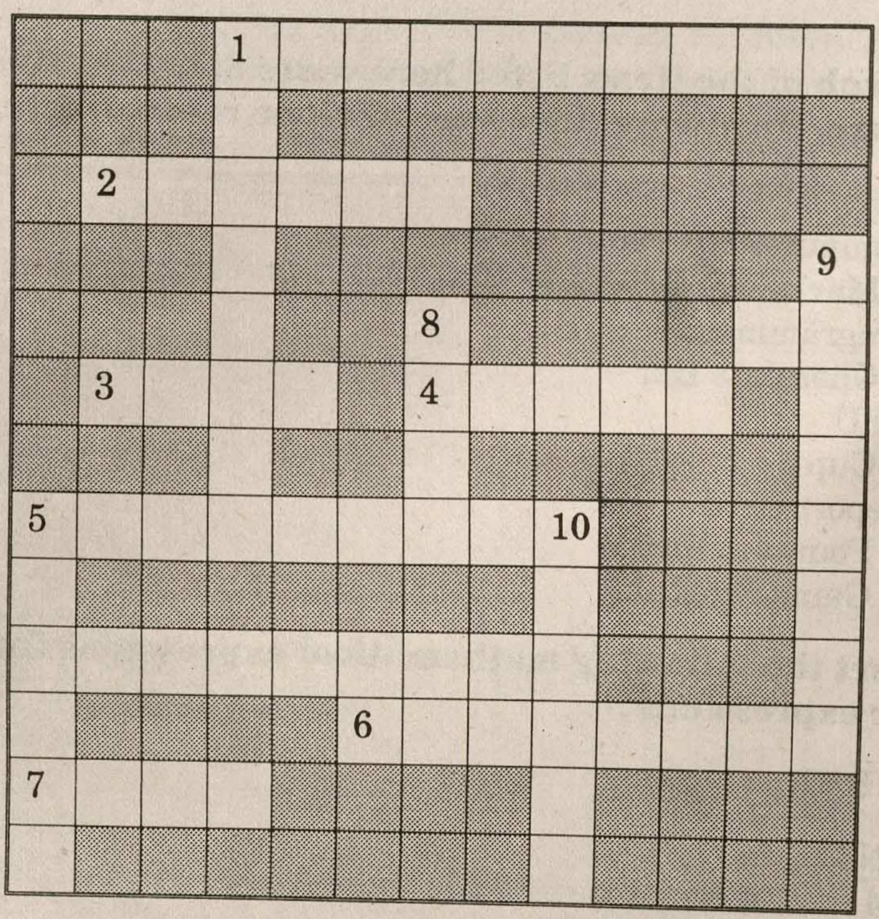
Q. IV. For each of the items listed here write S, L, N against it, if it is a software, Hardware, Liveware or None respectively.

1. Keyboard
2. Computer Operator
3. A Marksheet Program
4. Programmer
5. A Chocolate Bar
6. CPU
7. A Cup
8. Report Card
9. A Fountain Pen
10. A Game Program

Q. V. Convert the following mathematical expressions into valid computer expressions :-

1. $A(B*C)$
2. $\frac{2N}{M}$
3. $Z(Y)^3$
4. $4X - 7Y$
5. $(A+B)^2 + D$
6. $X^2 + Y^2$
7. $\frac{L+M}{J+K}$
8. $(T-Q)R + PT$
9. $(X-Y)(A+B)$
10. $\frac{E^2 + F^3}{G}$

Q. VI. CROSS WORD



Across

1. A device which performs simple calculations (10 letters)
2. Store room of the computer (6 letters)
3. Raw material of information (4 letters)
4. That which is fed in (5 letters)
5. Symbols which perform operations (9 letters)
6. Set of programs (8 letters)
7. The person who deals with the computer (4 letters)

Down

1. Device that accept manipulates and outputs information (8 letters)
5. The result (5 letters)
8. Wrong information, wrong output (4 letters)
9. Physical components of computer (8 letters)
10. Grammer of computers (6 letters)

2

BASIC

A letter can be written in any language provided the sender and receiver is able to follow it e.g. English, Hindi, Urdu, Punjabi etc.

Similarly to make a computer do things you need to give it instructions in a language it understands. These languages resemble common English and are called high level languages. The example of computer languages are as follows :

BASIC
COBOL
FORTRAN
PASCAL
LOTUS 1-2-3 etc.

In order to have effective communication with a computer, only a limited number of words may be used as predefined commands. Moreover, statements(sentences) that specify clear instructions to a computer must obey strict rules(grammatical) called syntax. This combination of restricted vocabulary & syntax is a programming language. BASIC is one such programming language.

BASIC is a language which is most elementary and simple, quite like English. The full form of BASIC is **B**eginners **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. BASIC's popularity is its widespread availability. It is fast becomming the "mother tongue" of modern computers.

You will very soon realise that programming in BASIC is not only easy but also fun.

It was first developed in 1964 by John Kemeny & Thomas Kurtz at Dartmouth College, New Hampshire and was intended to be ideal for a beginner.

Type of Characters

The character set refers to all the units of things put into different groups which can be used for a particular computer language. Character set for BASIC is as follows:

1. Alphabetic (Letters) Characters – A to Z (26)
2. Numeric (Digits) Characters – 0 to 9 (10)
3. Special Characters – The symbols other than letters and numbers e.g.:- , + ; ? $: etc.
   Space or blank is also a special character.

NOTE : You cannot use any character other than the above mentioned characters in your BASIC program e.g. अ, ਹ etc.. cannot be used.

Data & its Types

Data are the raw facts or figures which are fed into the computer to be processed. They are mainly of 2 types :-

Numeric : It is represented by numeric characters such as 3, 27 etc.

String or Alphanumeric :- It means either an alphabetic special character or a combination of these. Even numbers can be combined with the alphabets to make an alphanumeric data e.g. :-
"SCHOOL", "Yes 5.00 a.m.", A3, ZZ2 etc.

The items of data are further classified under the following heads:-

1. **Constants** :- Those data items whose value, remains unchanged during the execution of program are called *constants*. These are of two types:- Numeric constant and String constant.

   a. **Numeric Constant** : All numbers itself are numeric constant. As 77 will never be equal to 7. Examples 3255, 2, -10, 2.33 etc.

   b. **String (alphanumeric Constant)** : Any thing written within quotation marks are string constant. As "YES" will never become "NO". Numbers can also be written within double quotation marks but they cannot be used for calculations. Size of characters present in the string can vary from 0 to 255 characters.

Example : "TODAY", "HELLO! CHILDREN", "SAFE COMPUTING", "365", "254" etc.

2. **Variables** : Those data items which may assume many different values during program execution are called variables. A variable has a name which does not change, and a value which can change. It is used for naming or representing a value. The first character should always be an alphabet that may be followed by letter, number etc. There are 2 types of variables namely :-
   Numeric variables and String variables.

   a. **Numeric Variables** :- It is used to represent only numbers. Numeric variables or its values can be used for mathematical calculations.
      Valid Examples :- Z1, A, XX, CD etc.

   b. **String (alphanumeric) Variables** :- It is used to represent a string (of numbers, alphabets) etc. Remember, it should always end with a dollar sign i.e ($). These variables cannot be used for mathematical calculations.

Valid Examples : A$, BX$, N5$ etc.

Thus :-

```
              L = 5
             /    \
Numeric variables  Numeric constant

              L$= "50 apples"
             /       |
String variable     String constant
```

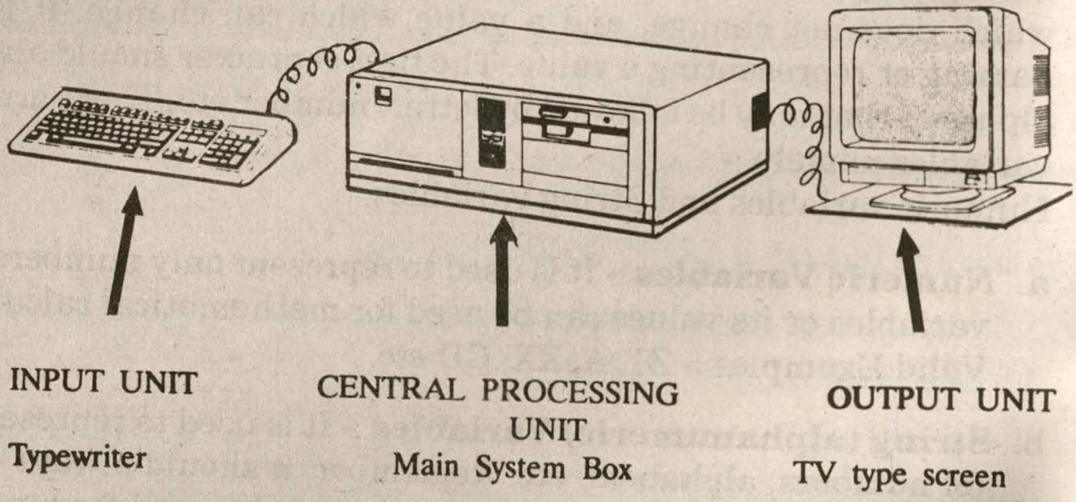
Expression :- A string of constants and variables each seperated by operators is an expression.
EXAMPLE : N+M, Y+6 - (Z ↑ 2) etc.

- You may have to use all the above mentioned things in building up your program.

Keyboard

Keyboard is an input device. It is a board which has several keys on it. It is like a typewriter and has keys similar to it. Computer has however some more special keys having specific functions. Each key has some symbol marked on it. The keyboard may differ in size, and by the number of keys.

The BBC micro computer has 74 keys in all. These keys are divided into 6 different groups depending on the function they perform.



1. **Alphabetic/Letters keys** - There are 26 letter keys. A–Z.
2. **Number keys** - They are 10 in number varying from 0–9.
3. **Symbol keys** - They are 12 in all. They consist of various symbols like +, -, /, *, ?, : etc.
4. **Function keys** - These keys are on the top row and are used for specific purposes i.e. user defined functions. They vary from F1 to F10 i.e. 10 in number.
5. **Edit keys** - They are used to make corrections i.e debug the program. They are 5 in all i.e. ↑, ↓, ←, →, copy.
6. **Control keys** - These are very important keys as they control the function of the whole computer. They help us to type and execute our program successfully on the computer. They are :
    a. **Escape** - To get rid of any work or to come out of any mode we press ESCAPE key.

    b. **Tab** - It is used to leave spaces.

    c. **Caps Lock** - It is used to change to capital letters from small letters and vice versa.

d. **Control Key** - It is always used in conjunction withother keys for specific function e.g. CTRL+G makes a sound.
CTRL+I moves a space to the right etc.

e. **Shift Lock** - Some keys of the keyboard have 2 marked functions. One is upper and the other is called lower function of the key. If we just press any key we get what is marked on the lower side of the key. For upper case function one has to press the shift lock and then press that particular key. To come out of this mode we will press shift lock again.

f. **Shift** - There are two keys for shift. Its function is the same as mentioned above but it does not lock or unlock. To get the upper case function of any key keep one finger on shift and the other on the required key.

g. **Return** - After typing a line to bring the cursor(–) to the next line we press RETURN.

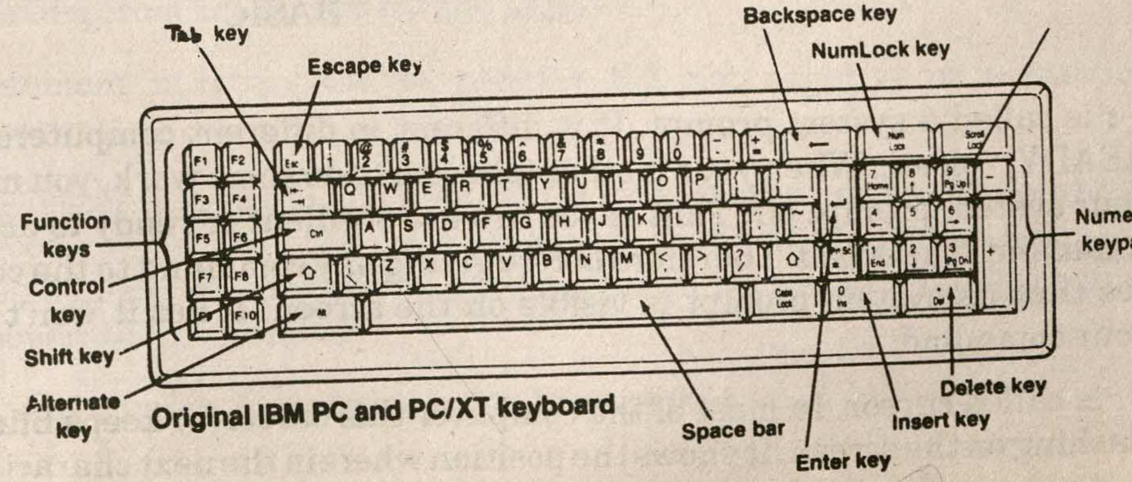
h. **Delete** - It is used to erase. Each time this key is pressed the cursor moves one space to the left and rubs out the character which was typed there and if we hold it on for a longer time it will clear the whole line.

i. **Space Bar** - It is used to leave spaces after typing every word.

j. **Break** - It clears the screen of the computer except the top 3 label lines. In a way, it resets the computer.

To know your keyboard better it is best to use it several times.

The IBM PC computer has 84 keys in all. Most of them are same as the BBC micro. The main difference is in the following keys:-



Original IBM PC and PC/XT keyboard

16843

1. **Enter** : It is the same as RETURN key as it moves the cursor to the next line.

2. **Escape** : It deletes the last line that the cursor is in.

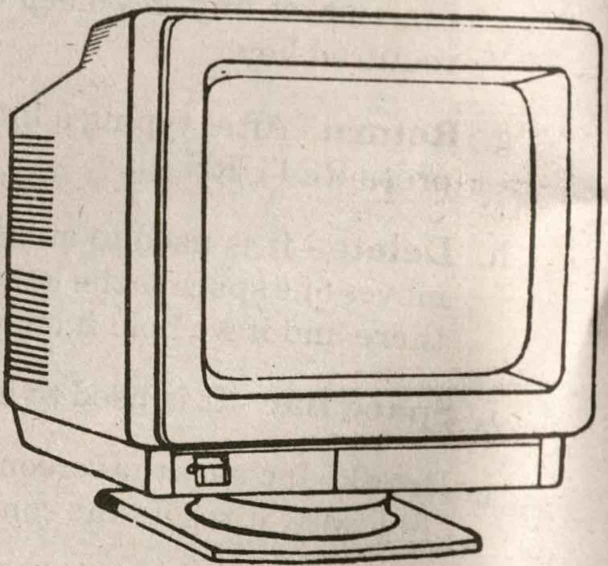
3. ← : It is equivalent to BS that is backspace. It is used to move the cursor one character to the, left thereby deleting the character.

4. **Scroll Lock** : When SCROLL LOCK is pressed with CONTROL key, it stops the program running in an infinite loop (non-stop). It is used to come out of any mode.

Monitor

This is also called Visual Display Unit (VDU). It is an output device, where in you can see your program (not Doordarshan programmes,) as well as its answer. It is just like your T.V. screen with all its control buttons.

On some BBC micro computers the screen will look like:-



```
BBC COMPUTER
32K ACORN DFS
BASIC
```

> : is called a system prompt. It is different in different computers i.e. A>, READY, ok etc. When your mother tells you to do some work, you may keep your system prompt off. That is you are not obedient or ready to accept her command. Thats bad. Therefore before giving any command to the computer see that its system prompt is visible on the screen. If not it won't execute your command.

– : is called cursor. In most of the computer this cursor(–) keeps blinking or flashing on the screen. It shows the position wherein the next character typed in by you will appear and then moves one place towards the right on the screen.

Points to Remember

Never hit the keys of the keyboard with a heavy hand, even a feathers touch will not do. The keys have to be pressed just right that is with a reasonable soft touch. Don't lower(you won't be able to see) or increase the brightness (to get a pair of extra eyes soon).

Press space bar before typing the next item.

Press Return (Enter) key after every command and after typing any sentence (statement).

Zero is seen as 0 whereas the alphabet 0 is shown as 0.

Line Numbers (statement Numbers) : In every book you must have seen a page number on every page. If the page numbers are not given, it would become very difficult to locate anything, which in turn would create confusion. Similarly the instruction to be given to the computer must be in an orderly manner with numbering. In other words every line of a Basic Program must have a line number given in the beginning.

2. You can give an instruction by using not more than 255 characters under one line number.
3. All line numbers should be numeric. It can even be 0.
4. The computer executes any program according to the line numbers starting from the lowest till the highest.
5. Statement numbers can be positive but not negative or a fractional number.
6. It is better to give the line numbers in an ascending order. Moreover give a good gap between the line numbers so that you can feed in some more instructions which you may have missed in between two lines by giving a seperate line number.
7. No line number in a program can be repeated.

8. It is not necessary that the gap between the line numbers of a program be equal. In other words, unequal class intervals between the line numbers are allowed .

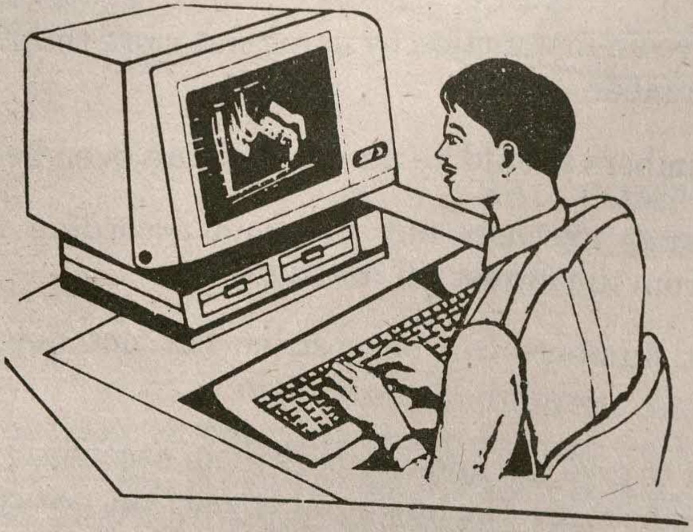
Valid Examples of Linenumbers :-

1	10	2	5	1
2	20	5	100	10
3	30	6	200	100
4	40	10	500	1000
5	50	28	900	10000

****

Be a Boss

I am sure, till now must have been under the command of different people, say Dad, Mum Teacher etc. Don't get disheartened, this is the time when you can become a boss by ruling over your computer. You, like a boss can give it instructions and the computer, like a good child will obey you. Yes, to get your work done you will have to instruct it. Following are the preliminary commands to make your program work successfully on the computer.



1. **NEW** : This command is typed before feeding any new program so that the previous program does not get mixed up with the present one.

2. **AUTO** : It means automatic. To get the line numbers automatically on the screen we type AUTO. It generates the statement numbers starting with 10 and in multiples of 10. That is :-

   10
   20
   30
   40 and so on.

3. **RUN** :- To get the answer of the program that has been typed RUN is used. In other words, RUN is used to execute any program.

4. **LIST** : To see a program again on the screen according to the line numbers LIST is used. To display a particular line of a program type LIST line number (where line number is any line number of the program which is to be displayed).

5. **CLS** : CLS stands for CLEAR SCREEN. To clean or wash off the screen thoroughly CLS is used.

6. **DELETE** : To rub or erase any line of a program type DELETE and give the specified line number.

7. **RENUMBER** : This command is used to rearrange or renumber all the lines of a program. It gives the first line the number 10 and all subsequent lines are numbered in increments of 10. In some cases if the user wants to modify his program, by inserting a line in between line numbers like 12 & 13, RENUMBER command becomes useful.

☞ REMEMBER

1. Always type the command in front of the system prompt.
2. Always type the command in capital letters.
3. After typing any command press RETURN(ENTER) key.
4. Dots or full stops/periods are not allowed in BASIC programs. It can only be used as decimal point for fractional numbers.
5. Linenumbers are not required to give the commands.

EXERCISE - 2

I. State whether the following statements are true or false :-

1. BREAK key is used to break the computer.
2. The computer displays system prompt A> or OK on the screen only when it is busy and does not like to be disturbed.
3. Line numbers in a basic program are important as they act as a pointer.
4. "LOVE" is a string constant.
5. CAPSLOCK key is used to lock the caps of the computer.
6. Numbers stored within quotation can be used for calculations.
7. END is a command.
8. PRINT is a statement.

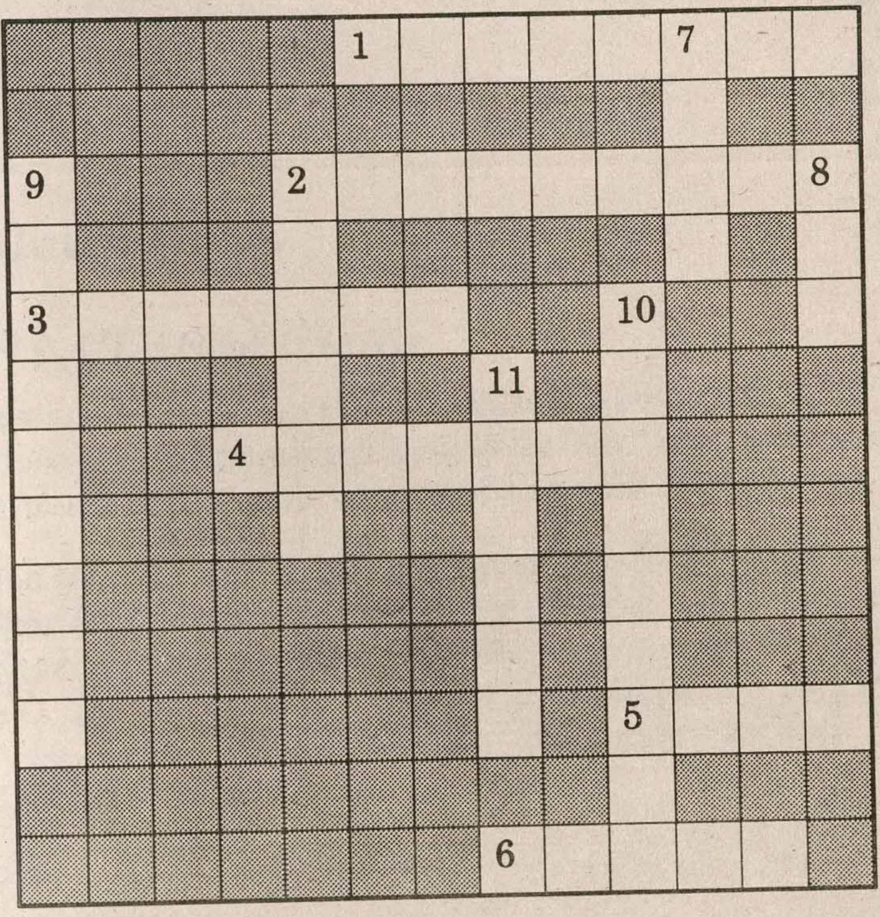
II. What is the function of the following commands.

1. AUTO
2. RUN
3. LIST
4. CLS

III. Name any two :-

1. Computer languages
2. Commands
3. Letter keys
4. Statements
5. Numeric constants.

IV. CROSS WORD



Across

1. Typewrriter like input device (8 letters)
2. A single unit (9 letters)
3. Term used for numbers (7 letters)
4. T.V. like screen (7 letters)
5. Command used to see the program again (5 letters)
6. Elementary language of computers (5 letters)

Down

7. Command used to get linenumbers (4 letters)
8. Command used to get the answer (3 letters)
9. Data items whose values remain unchanged (9letters)
10. Data items whose values change (9letters)
11. Another name for alphanumeric characters (6 letters)

11,110

3

Simple Calculations

USING PRINT, CLS Statements

Lady Ada lovelace, Lord Byron's daughter, is known to be the "first programmer". Therefore a programming language called "Ada" is named after her. This chapter deals with simple computations with the help of a computer.

If you wish to add 7 and 5. You can't just type 7+5 or type add 7 to 5 and get the answer as the computer won't accept it or will call it a mistake. It has to be written in a particular format i.e. program. Although simple short calculations (1 line calculations) need not be written in a program form but remember to type PRINT before the calculations and press RETURN(ENTER) after it.

For example

```
>PRINT     700
  700

>PRINT     5+6
  11

>PRINT     5-6
  1

>PRINT     5*4
  20

>PRINT     50/5
  10

>PRINT     5 ^ 2
  25
>PRINT     11+6-4*3
  5
```

```
>PRINT     (6/3 + 12/4) / 2
2.5
```

To display an alphabet or message on the screen it should be enclosed within quotes.

Example:

```
>PRINT     "T"
 T

>PRINT "COMPUTERS ARE FUN"
 COMPUTERS ARE FUN

>PRINT "5.00 A.M."
 5.00 A.M.
```

Similarly

```
>PRINT "GOD-IS-GREAT"
 GOD-IS-GREAT

>PRINT "3+2"
 3+2

>PRINT "Yes"
 Yes
```

If you wish to print more than one item by using one PRINT statement then don't forget to use semicolon or comma between the items.

a. If comma is used the computer will leave some spaces before printing the next item. Thus the items will be more widely spaced out.

b. If semicolon is used the data items will be displayed just next to each other with no space left between them.

c. If hypen is used the next item will be printed on the next line. (Only in case of BBC micro computer).

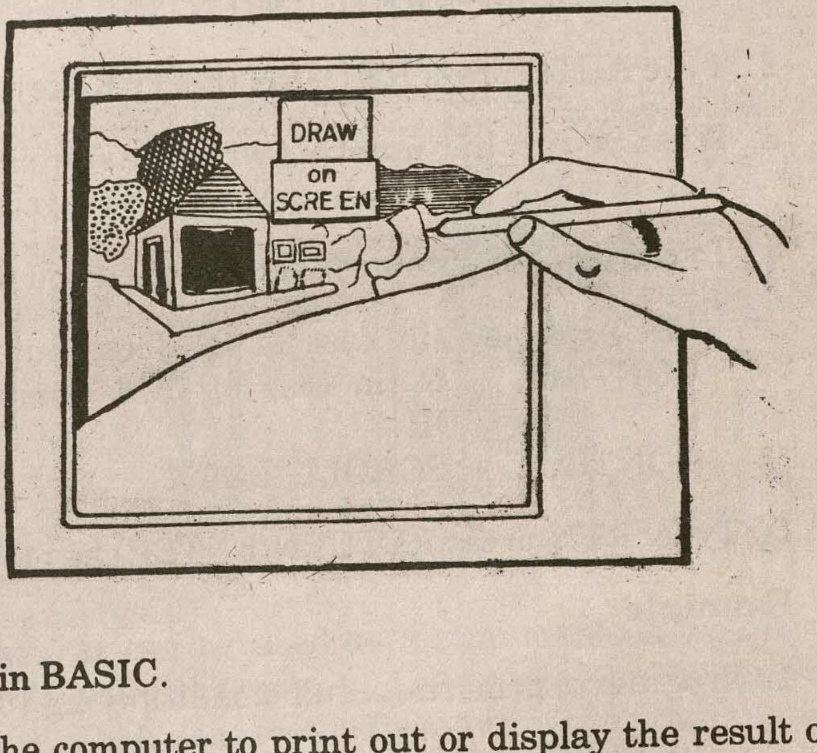
For example

```
>PRINT 30,20
30         20
```

```
>PRINT 30;20
3020

>PRINT 30’20
30
20

>PRINT “A”,“B”;“C”
A      BC

>PRINT “RIGHT”,“YES”,“O.K.”
RIGHT YES
O.K.
```

A program is a group of instructions or numbered lines called statements. A statement simply states something or ask the computer to obey something. Hence, those statements that do things are called executable statements. The examples of executables statements are : **PRINT, LET, END, GOTO** etc.

PRINT Statement



1. PRINT is a statement in BASIC.
2. It is used to instruct the computer to print out or display the result or information on the screen.
3. It allows, the computer to communicate with the user by outputting information on the VDU.

4. It can be used to print all kinds of expressions,variables,constants and messages.
5. The items to be printed should be written after PRINT and should be seperated by a comma or semicolon.
6. It causes the printing to start on a new line that is each print statement prints on a seperate line.
7. The format or syntax of PRINT is :-

   1 PRINT list
   where **1** stands for line number
   list is the list of items to be printed

If you have a long calculation or if your calculation needs more then one line then you have to write it in a proper program form.

It means you have to :-

1. Type NEW press RETURN (ENTER) key.
2. Type AUTO press RETURN (ENTER) key.
3. When you get line numbers write the statements and then the calculation, that is when you finish up with the program, press RETURN.
4. To come out of automode that is when you finish up with program, press:

   ESCAPE key (in case of BBC computer)
   CONTROL + C (in case of IBM Computer)
   OR
   CONTROL + SCROLL LOCK
5. Type RUN press RETURN(ENTER) key to get the answer.

Example

1. Program to print the result of multiplying 10 by 3 and adding 50 and 100.

```
10 PRINT 10*3
20 PRINT 50+100
30 END
```

2. Program to print your name and age.

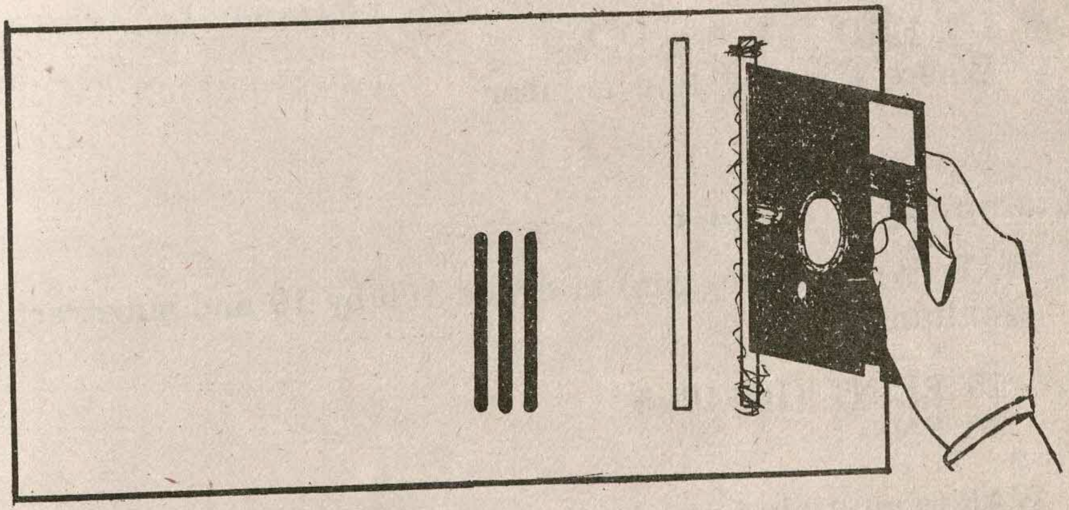
```
10 PRINT "Master XYZ"
20 PRINT "5 YEARS"
30 END
```

In the absence of a list (of items) a blank line is printed.

For example

```
10 PRINT "I LOVE"
20 PRINT
30 PRINT "COMPUTER"
40 END

RUN
I LOVE

COMPUTER
```

CLS Statement



CLS stands for clear Screen. It can be used as a statement as well as a command. Its function, as the word suggests is to clean or clear the screen. When used as a statement that is if used on the first line of the program the output of that program will appear on a blank screen.

FORMAT : 1 CLS
where 1 is any line number.

Program

```
10 CLS
20 PRINT "HONESTY IS THE"
30 PRINT "BEST POLICY"
40 END
```

END Statement

1. It is a statement in Basic.
2. It may be the last statement of any program.
3. It is better to assign some highest line number to END statement.
4. Once the computer meets the END statement it stops the execution of the program.
5. It shows executional as well as the physical end of the program. In other words, it acts as the point of termination of the whole program.
6. Format :

```
l    END
```

Where l is any line number

Examplory Programs

1. WAP (Write a program) to divide 100 by 10 and substract 5 from the resultant.

```
10 PRINT (100/10)-5
20 END
```

2. WAP to multiply 5 numbers.

```
10 PRINT 8*3*15*6*2
20 END
```

3. WAP to calculate the cost of 50 cricket bats when one bat cost Rs 60/-.

```
10 PRINT 60*50
20 END
```

4. WAP to print “Computer” 5 times on the screen

```
10 PRINT “Computer”
20 PRINT “Computer”
30 PRINT “Computer”
40 PRINT “Computer”
50 PRINT “Computer”
60 END
```

5. WAP to find the area of a square chocolate box when its one side is equal to 10 cms.
[Area = side*side or side^2]

```
10 PRINT 10  ^ 2                  10 PRINT 10*10
20 END              OR            20 END
```

6. WAP to print a square.

```
10 PRINT “* * *”
20 PRINT “* b̸ *”     [ b̸ means press the space bar once]
30 PRINT “* * *”
```

7. WAP to print (calculate) 20% of Rs. 120/-.

```
10 PRINT (20/100)*120
20 END
```

EXERCISE – 3

I. DRY RUNS : What will be the output of the following programs:-

1.
```
10 PRINT 230,11000
20 PRINT 230+11000
30 END
```

2.
```
10 PRINT "JACK & JILL"
20 PRINT
30 PRINT "WENT UP THE HILL"
40 END
```

3.
```
10 PRINT 7; 777
20 PRINT 7-777
30 PRINT 7*777
40 END
```

4.
```
10 PRINT "Basic", "is"
20 PRINT "just", "Basic"
30 END
```

5.
```
10 PRINT 120+40
20 PRINT 120-40
30 PRINT 120*40
40 PRINT 120/40
50 END
```

6.
```
10 PRINT "HELLO!"
20 PRINT
30 PRINT
40 PRINT "Mr. COMPUTER"
50 END
```

7.
```
10 PRINT "2+2"
20 PRINT "makes"
30 PRINT 2+2
40 PRINT "RIGHT"
50 END
```

8.

```
10 PRINT 6+9; 6-9;6*6;9*9
20 PRINT 6 9, 9 2,6/9
30 END
```

9.

```
100 PRINT (100/100)*100
200 END
```

10.

```
10 PRINT 5;“*”;“4;=”;5*4
20 END
```

11.

```
10 PRINT “V”,2
20 PRINT “R”,1
30 END
```

12.

```
10 PRINT “U”,“CAN B A”
20 PRINT “MASTER OF”; “COMPUTERS”
30 END
```

13.

```
10 REM PRESS SPACE BAR AS MANY
20 REM TIMES AS b̸ APPEARS
30 PRINT “*****”
40 PRINT “b̸b̸*”
50 PRINT “b̸b̸*”
60 PRINT “b̸b̸*”
70 PRINT “b̸b̸*”
80 END
```

14.

```
10 CLS
20 PRINT “----TIME-TABLE”
30 PRINT “DAY”, “SNACKS”
40 PRINT “MONDAY”, “POP CORNS”
50 PRINT “TUESDAY”, “CHOPS”
60 PRINT “WED”, “CHOCOLATES”
70 PRINT “THOU”, “BURGERS”
80 PRINT “FRI”, “PASTRIES”
90 PRINT “SAT”, “PATTIES”
100 PRINT “SUN”, “ICE-CREAM”
110 PRINT “*THANK YOU****”
120 END
```

```
15. 100 CLS
    200 PRINT "- - - - -"
    300 PRINT "b̸b̸ - b̸b̸"
    400 PRINT "b̸b̸ - b̸b̸"
    500 PRINT "b̸b̸ - b̸b̸"
    600 PRINT "b̸b̸ - b̸b̸"
```

II. Debugging : Correct the following programs:-

```
1.  10 PRINT 7+3-2
    20 END
```

```
2.  10 PRINT (5+6*9)8
    20 PRINT "DEBUG"
    30 END
```

```
3.  10 PRINT 10 × 100 × 1000
    20 PRINT RS 5
    30 END
```

```
4.  100 PRINT "PRINT"
    120 PRINTS "PRINT"
    130 WRITE   "PRINT"
    140 END
```

```
5.  10 PRINT STATEMENT
    20 PRINT "&"
    30 PRINT "COMMAND"
    END
```

```
6.  100 PRINT "GOOD"; MORNING
    200 PRINT "SALAAM"
    300 PRINT "NAMASKAR"
    400 THE END
```

```
7.  10 CLTS
    20 PRINT "I LOVE"
    30
    40 PRINT "INDIA"
    50 END
```

8.
```
1.1 PRINT    ******
1.2            ***
1.3             *
1.4 END
```

III. Fill in the blanks according to the specification:-

1.
```
10 CLS
20 PRINT "100", 100+200
30 ----
```

2.
```
10 __________
20 PRINT "7 SEAS"
__ END
```
[The output should appear on a cleared screen]

3.
```
10 PRINT "DELHI" ____ "IS"
20 ___  "THE CAPITAL"
30 PRINT "OF INDIA_
40 END
```

4.
```
10 PRINT 11_22_33_44
20 END
```
[The output should be the product of 4 numbers]

5.
```
100 PRINT 100_____2
— _____
```
[The output should be the square of 100]

IV. Programs :-

1. WAP to add 700,70 and 7.
2. WAP to multiply 300 by 3000 and divide 40 by 4.
3. WAP to find the price of a pen when 100 pens cost Rs. 950/-.
4. WAP to print your name four times on the same line of the screen.
5. WAP to print the name of your school followed by its address.
6. Mr. XYZ pays 15% of his salary as house rent. His salary is Rs. 2,800. WAP to calculate how much does he pays as a house rent.
7. WAP to find the area of a square which is 70 mts long.
8. Mr. R took a hundred rupee note to a shop. He bought tea powder for Rs. 18.50, sugar for Rs.20/- and patties worth Rs. 25/-. WAP to calculate how

much money did he spend and how much remained with him after the purchase.

9. WAP to print the total cost of 10 items at Rs.53/- each & 92 items at 80 paise each.
10. A man wants to buy 4 barss of silver to make an offering. One such bar weights 5.55kg. WAP to find the total weight.
11. In a computer test 640 students appeared out of which 40 failed. WAP to calculate how many candidate passed.
12. WAP to print a rectangle:-

```
%%%%%%%%%%%%
%          %
%%%%%%%%%%%%
```

[WAP stands for write a program]

13. WAP that outputs the national flag?

4

LET and REM Statement

In this chapter, you shall learn about the statement that makes the computer accept information and display remarks on the screen.

LET statement

1. LET is a statement in BASIC. It is just like saying suppose. For example:- Suppose the price of badambar is Rs.4/- Baby Richa wants to buy 10 badambars. So how much will she pay?

General calculation
1 badambar = Rs. 4/-
so 10 badambar = Rs. 4*10
= Rs. 40/- That is She will have to pay Rs40/-.

Similarly, in Basic language the same problem can be tackled as:-

```
10 LET PRICE = 4
20 LET NUMBER = 10
30 LET COST = PRICE*NUMBER
40 PRINT COST
50 END
```

When the computer reads line number 10 it understands the instruction "LET PRICE" as "Open up a memory box named by the variable name "PRICE".

Next part of the instruction "=4" as "Fill in the numeric constant 4 in the memory box "PRICE".

Right, in the same way, it will store 10 in another memory box named "NUMBER".

In order to find the cost of the item, we need to find the product of the contents of memory box PRICE and the contents of the memory box NUMBER. The total cost that we get will be stored yet in another memory box named by another variable name "COST".

Finally the contents of the memory box "COST" are displayed.

2. LET statement is used to assign a value to a variable. It is also called an assignment statement.
3. It places the given value into a named memory location.
4. LET statement is optional in most of the computers. In BBC micro computers and IBM PC it is optional to use LET.
5. Format(Syntax)

```
1 LET v = a
where 1 = any line number
v = any variable name
a = any constant any variable or any expression
"=" is called assignment symbol.
```

6. LET statement can be used to assign values to numeric variables.

```
a.  10 LET M = 10
b.  200 LET XY = 3700
c.  60 LET Z1 = 1
d.  266 LET T = 825
```

7. LET statement not only assigns values to numeric variable but also to string variable.

```
a.  30 LET C$ = "INDIA"
b.  70 LET L$ = "THE*BEST*"
c.  36 LET XY$ = "COME ON, BACKUP COMPUTER!"
d.  14 LET J5$ = "GOD IS GREAT"
```

8. Last but not the least it represents an expression or a calculation.

```
a.  100 LET B = 36+3+6
b.  16 LET NN = (A+B)/C
c.  20 LET KM = F*E-G+H
d.  200 LET Q1 = (P+Q)/8
```

Do & Don’t (Remember)

1. As mentioned earlier in many versions of BASIC, LET is optional i.e. you can write in the following way.

```
i)   10 E = A+B
ii)  20 X = F+32*C
```

2. You can <u>never</u> use LET to assign variable to value

```
i)   99 LET “GOOD” = G$
ii)  20 LET X+Y+Z = R
```

3. LET statement can be used to give more than one values to the same variable name within a program.

```
i)   10 LET N=5
     20 LET N=N*N
     30 PRINT N
     40 END

     RUN
     25

ii)  100 LET Z = 2
     110 PRINT Z
     120 LET Z = 6
     130 PRINT Z
     140 END

     RUN
     2
     6
```

4. When you assign values to variable, which is a string, it is better to avoid a text having quotation marks because some computers may consider it as the end of the string, instead you may use apostrophe in a string.

```
i)  20 LET K$ = “RAM SAID,“THAT’S BAD””
ii) 77 LET MY$ = “THAT’S IT”
```

5. It is possible to assign a number value to a string variable but remember it cannot be used for calculations.

```
i)   35 LET TO$ = "2"
ii)  350 LET J = TO$*7
```

6. When one variable is made equal to the other variable. The first variable looses its original value(if any)and gets the value of the later.

```
i)   110 LET A = B
```

(Whatever is the value of B becomes the value of A as well).

REM Statement

1. It is an unexecutable statement in Basic.
2. REM stands for REMARK.
3. As the name suggests it is used to place comments or remarks in the program.
4. It is ignored by the computer.
5. REM statements acts as a reminder to the programmer or whosoever reads the program to make him understand it better.
6. It can be used anywhere and as many times as required in a program. As it is a nonexecutable statement therefore it can even be written after the END statement.
7. Format :-

```
1 REM c
where  1 = any line number
       c = any comment
```

8. Thus, REM statements show comments about the operations in the program.

Example :-

```
i)   10 REM This is a basic program
ii)  220 REM Programmer : Miss computers
iii) 12 REM Date written 30/4/89
iv)  90 REM  PROGRAM TO ADD FOUR NUMBERS
```

Use of Colon

1. In a Basic program it is possible to put more than one statement using a single linenumber so long as the individual statements are seperated by colons(:).
2. We can, with one linenumber write as many statements as possible so long as the line has less than 230 characters. But it differs from computer to computer.
3. Its great advantage is that it saves memory space and sometimes may make the program work a bit faster.
4. Its disadvantage is that it is a bit difficult to make any corrections or to follow (understand) the program when you list it.

Example :-

i)
```
10 LET X = 30 : LET Y = 40 : LET Z = X+Y : PRINT Z : END

RUN
70
```

ii)
```
40 PRINT "JACK&JILL":PRINT:PRINT"WENT UP THE
HILL":END

RUN
JACK & JILL

WENT UP THE HILL
```

Examplory Programs : [Using LET, REM & colons]

i)
```
10 CLS
20 REM DEMONSTRATION
30 REM **************
40 LET D$ = "Programming is Fun"
50 LET K = 420
60 PRINT D$,K+K: PRINT "YES"
70 END
80 REM ************
```

ii)
```
10 REM %%%%%%%%%%
20 REM PRINTING
30 LET T = 4:LET U = 8:LET V = 12
40 PRINT T,U,V
50 REM  ----------
60 PRINT T;U;V
70 END
80 REM *THE *END.
```

EXERCISE – 4

I. DRY RUNS : What will be the output of the following programs :-

1.
```
10 LET P = 25
20 PRINT P,P;P
30 END
```

2.
```
110 CLS
120 LET K = 80
130 PRINT K:PRINT:PRINT K+K
140 END
```

3.
```
10 LET M$ = "MR. COMPUTER"
20 LET N$ = "YOU ARE GENEIOUS"
30 PRINT M$,N$
40 END
```

4.
```
10 LET L = 10 :PRINT L-L:PRINT L+L:PRINT L*L:
PRINT L/L:PRINT L↑2:END
```

5.
```
100 LET T$ = "TREAT": LET Y$ = "YOU"
200 LET O$ = "ALWAYS OTHERS AS"
300 LET W$ = "WOULD LIKE THEM TO"
400 PRINT T$,O$ : PRINT Y$,W$
500 PRINT T$,Y$:END:REM **
```

6.
```
1 REM DOUBLE ROLE *
2 REM PROGRAM *
3 LET Q = 8
4 PRINT Q*5,"YES"
5 LET Q = 10
6 PRINT Q*Q+Q, "NO"
7 END
```

7.
```
10 LET R$ = "**"
20 PRINT R$
30 PRINT R$,R$
40 PRINT R$ : END
```

8.
```
10 PRINT "*N-*-*"
20 REM "*-*-*"
```

```
30 LET A = 5
40 PRINT “A”,A,A*A
50 PRINT A 2,A 3,A-A
60 END
70 REM “*-*-*”
```

9

```
10 PRINT “100+100 = 200”
20 PRINT “PRINT”
30 REM PROGRAMMER MR.P.
40 PRINT 100,100,200
50 PRINT “U AND E”
60 REM *OVER*
70 END
```

10.

```
10 LET A=5: LET B = 60
20 PRINT A+B,A-B
30 PRINT A*B,B/A
40 PRINT A;B
50 PRINT B,A
60 END
```

Debugging : Correct the following programs:-

1.

```
10 REM ** DIVIDE TWO NUMBERS **
20 LET * = 64; LET Y = 7
30 LET X/Y
40 END
```

2.

```
10 REM * TO PRINT THE MESSAGE*
20 LET K = “WORRY GIVES SMALL THINGS”
30 LET J$ = BIG SHADOWS
40 PRINT K$,J$
50 END
```

3.

```
10 LET B$ = “100”
20 LET E = B$*B$
30 PRINT E
40 END
```

4.

```
10 REM* TOTAL MARKS OF 4 SUBJECTS*
20 LET B = 80: LET F = 90: LET P = 60 : LET C = 70
```

```
30 LET B+F+P+D
40 END
```

5.
```
10 LET “MOTHER” = M$
20 REMARK -------
30 PRINT M$
40 PRINT MAA
50 END
```

6.
```
10 PRINT 10
20 LET G$ = “*’S”
30 PRINT G$, IN THE
40 PRINT “SKY”
50 ENDED
60 REM*REMARK*
```

7.
```
10 REM-AREA OF A SQUARE
20 LET SD = 70
30 PRINT D*S
40 END
```

8.
```
100 LET X = “ABC”
200 LET Y$ = “OF COMPUTER”
300 LET Z$ = 1989
400 PRINT X$,Y$
500 PRINT Z$
600 END
```

9.
```
10 LET L$ = “LADY”
20 LET FINGER = F$
30 PRINT L$:F$
40 END
```

10.
```
100 LET R = 6
20 LET S = 7
300 LET T = 8
400 PRINT RST
500 REM END
```

III. Rewrite the following statement by filling in the blanks :-

1.
```
10 CLS
20 LET M$ = "SCHOOL"
30 ---------
40 END
```
[The output to be value of M$ to be printed twice on the screen]

2.
```
10 _____
20 LET N$ = "_____"
30 PRINT _____
40 PRINT _____
50 PRINT _____
60 END
```
[To print your name three times on the cleared screen.]

3.
```
10 LET X=____: LET ____ = 30
20 ____ Z=X*Y :_____ Z :END
```

4.
```
10 _____CP=_____
20 _____PR=SP____CP:PRINT_____
30 END
```
[To calculate the amount of profit on sale of item cost in Rs450/- but sold for Rs 540/-.]

5.
```
10 _____A = 99 : LET B = 79 : LET ___ = 89
20 LET D = A+B+C
30 PRINT D–3
40 END
```
[To calculate the average of 3 numbers]

IV. Match the following :-

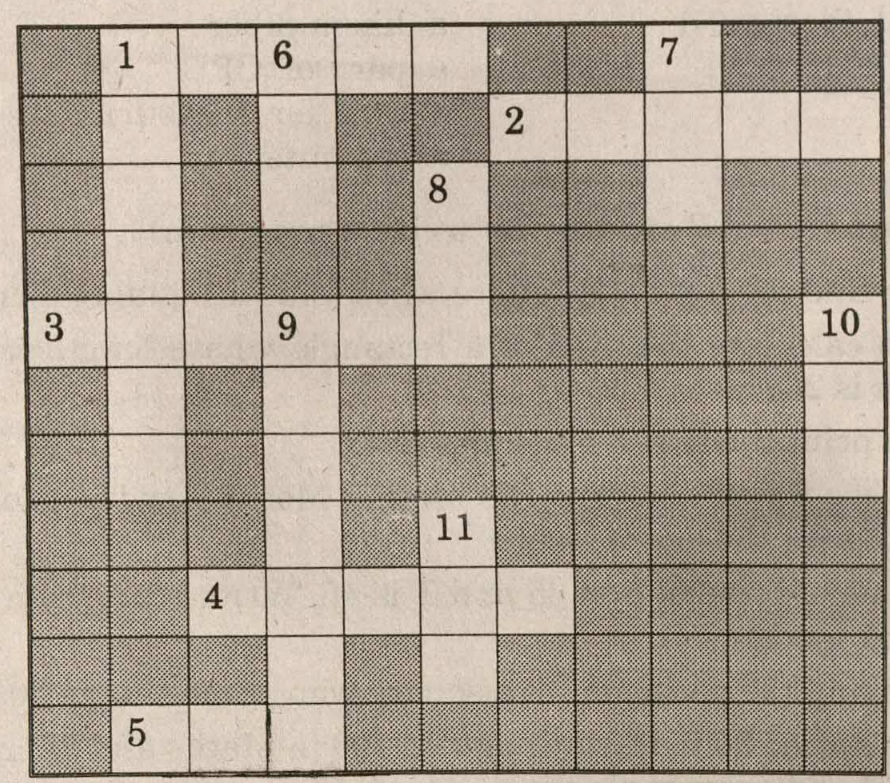
1.	PRINT	a mathematical operator
2.	PRINTER	a programming language
3.	REM	a nonexecutable statement
4.	NEW	a part of liveware
5.	KEYBOARD	an executable statement
6.	↑ or ^	an input device
7.	PROGRAMMER	a command
8.	BASIC	an output device
9.	MM$	a character
10.	U	a relational operator

11.	CONTROL UNIT	string variable
12.	K	a string constant
13.	Equal to sign(=)	a linenumber
14.	"TIT FOR TAT"	a part of CPU
15.	20	a unit for measuring the power of a computer.

V. Programs : (WAP stands for write a program)

1. WAP to find the area of a square whose side is equal to 4 cms.
2. WAP to calculate the area of a rectangle whose length is 10 cms and breadth is 20 cms.
3. WAP to print the first 3 multiples of 4.
4. Chottu is a 6 years old boy. His brother Mottu is twice as old as Chottu. WAP to print the age of Mottu.
5. The value of R is 25, S is 35 and T is 45. Write a program to find their sum and product.
6. WAP to find the perimeter of a square whose one side is 200 cms.
7. Qctavia scored 90 marks in English, 100 in Maths and 85 in Hindi. Find her total and average marks.
8. A maruti car travels at a rate of 60 kms per hour for 8 hours. Calculate and print the distance the car has travelled.
9. Mr. Retailer bought several articles for Rs 2000/- which he sold by making a profit of 20%. Calculate & print in a program form cost at which they were sold.
10. Mr. Eggbest sold 60 eggs for Rs 40/-. WAP to print the price of eggs per dozen.

VI Cross Word



Across

1. To correct an error (5 letters)
2. Used to seperate two or more statements using a single linenumber (5 letters)
3. Result (6 letters)
4. Statement used to display results (5 letters)
5. International business machines (3 letters)

Down

1. Process of typing the program and getting output (7 letters)
6. An error (3 letters)
7. Clear Screen (3 letters)
8. An assignment statement (3 letters)
9. Set of instructions given to a computer (7 letters)
10. Statement to give comments (3 letters)
11. Statement to terminate the program (3 letters)

5

Corrections of Bugs

You must have heard people saying "No man is perfect" and so it is true. When you type out several programs on the computer you may come across many different kinds of errors. Sometimes you get funny messages for your mistakes. It seems there won't be an end to it. Friends, don't get discouraged. There is always a way out.

Any bug can be debugged in several ways while typing any program if you make a mistake,you can rectify it by the use of delete[Back space or BS ←] key.

Whereas if you realise your mistakes ,after typing out your program you can debug it by any of the few methods discussed here.

1. To delete a line :- After typing out the following program if you feel that line 20 is unwanted and you want to get rid of it. Then when you get the system prompt, on the screen just type the required linenumber & press RETURN(ENTER) key. To check whether line 20 is deleted from the program type LIST & press (ENTER) key you will find only linenumbers 10 &30 with its contents.

Program-1

```
10 PRINT 7+5
20 PRINT 7
30 END
>20                 R
>LIST               R
10 PRINT 7+5
30 END
```

2. To insert a line :- After typing out program 2 as above, if you wish to add one more line in between 20 & 30. Or if by mistake you miss out a line you can always make an insertion. In front of the system prompt, type any line number (a number from 20 to 30 say 22,25,27 etc.) and type out the required matter in front of it but don't forget to press RETURN (ENTER) key. To check whether the insertion is being done LIST out the program.

Program-2

```
>NEW             [R]
>AUTO            [R]
10 PRINT 7+5     [R]
20 PRINT 7       [R]
30 END           [R]
ESCAPE
>26 PRINT 5      [R]
>LIST            [R]
10 PRINT 7+5
20 PRINT 7
26 PRINT 5
30 END
```

3. **To correct a line** :- After typing out PROGRAM-3 by mistake I have written PRINT as PRNT. When the system prompt appears type out the same linenumber in which you made a mistake & retype the contents of that line correctly. Press ENTER(RETURN) key. If you wish to check then LIST out your program. In this way the old line is replaced by the new one.

Program-3

```
>NEW             [R]
>AUTO            [R]
10 PRNT “A”      [R]
20 END           [R]
ESCAPE
>10 PRINT “A”    [R]
>LIST            [R]
10 PRINT “A”
20 END
```

6

Flowcharting

Development is an on going process. This happens in every sphere of life. Be it in science & technology, or in human's quest for better living. Better living also entails construction of new sites for homes and offices. Now-a-days, land development is everywhere going at a faster pace. Every piece of bizarre land sprawls up in an impressive office complex or a beautiful multistoreyed flat.

In case you have to visit one of these newly built localities, finding the correct place is often a difficult task. To simplify your approach, generally detailed maps purposefully indicate the location of the pillar or post where itself this map is situated and denote them as "You are standing here". Now it entirely rests on you to find the shortest route to your destination. Here you may want to remember peculiar identification points enroute to simplify your approach.

If you consider programming the computer as a goal (destination), you have to identify a definite path with a formulated strategy. This strategy will have a starting point, place to decide where to change the direction and the check points to confirm your approach. Only after sufficient experience, you can divide a strategy that is less time consuming and fast in solving the problem.

Flowcharts are basically diagrams drawn to finalise the approach to actual programming. In other words they merely represent the approach in terms of special symbols. Once put on paper, they become an easy reference to visualise the thinking of the programmer, without actually converting the thoughts to instructions. Remember, flowchart as such cannot be fed into the computer.

Consider another example. While studying, say in the night, you want to make a cup of Tea. Here the ingredients are water, tea leaves, sugar, milk and other necessary items such as a filter, a utensil, a gas stove and finally a cup to hold the tea. You definitely know the procedure to be followed.

1. Ignite the match stick
2. Light the stove
3. Put the utensil on the stove
4. Pour water in required amount
5. Wait till the water boils
6. Add sugar
7. Add tea leaves
8. Wait till the leaves colour
9. Add milk
10. Wait till a boil comes
11. Filter the tea into a cup

You must be wondering as to why I have mentioned all this? Well, I want to underline the fact, that even if the procedure is fairly simple, it depends on many if's and but's. In general, we can say that the main procedure is of boiling water and later adding sugar, tea leaves and milk. Rest is just the routine.

Well, yes! but, consider some of the if's and but's:

1. The gas in the stove might not be sufficient to allow carry on the process.
2. The match stick may get damp, failing to light. Even if everything else is in order, this small little thing might spoil the show.
3. The milk may sour the moment you add it to water.
4. The proportion of water and milk and tea leaves may not be proper resulting in a poor-tasting tea.

The moral!

Sometimes this may not happen as the way you presume. It may be possible that you had deviced the strategy as per your convenience, while practically that may not be feasible. Or more subtly, another obstruction might crop up the moment you deviate from the normal practice.

While visualizing a problem it may happen that some steps might be skipped because we assume these steps to have happened automatically, but they do not.

Suppose, you are locating a friend's house. And you miss a particular street, and by chance, you approach to a dead-end. Here, normally you might turn back and find the street you missed. Here "turning back" will be done automatically, where as the computer will have to be instructed to "turn back".

Consider another question:

We have mentioned 11 small steps for preparing tea. You may want to shorten or eliminate some steps or combine them in one. Say like instead of adding sugar, at one instance. Or you can mix everything and put it on the stove.

Agreed! Yes, you can adopt this method, but in this case, the tea might not taste as predicted.

Remember, you can alter one or several steps, but there are some procedures that have to be followed in a particular order. You have to ignite the match stick first and then the stove. But you cannot reverse the process. Similarly, only after the tea is prepared, you can use the strainer to filter it. Doing this earlier has no meaning. Take the case with our buddy, Computer. You must be aware that it is a dumb fellow. Everything, each and every step has to be told clearly before it can do the task given to him successfully& quickly. For you to program,some proper path has to be followed. Some strategy is to be drawn, some kind of rehearsals conducted to finalise the approach. But this is to be done without the help of computer. This path for conducting rehearsals is put on paper in the form called Flow-charts.

You are having your examination. The time is running out and you still have to answer some questions. Your goal (at minimum) is to achieve 60 percent of marks. Suddenly you realise that you committed a mistake in your current numerical and the answer is incorrect. Here you have to make a decision.

1. Recheck the numerical from the beginning. (But this will gobble a lot of time.)
2. Start attempting another numerical. (You already have committed a mistake in the previous example, so you have to take utmost care not to commit another one here. So expenditure on time will be on the higher side)

Decision has to be taken. Either recheck the numerical and ensure the marks or try the next one.

Similarly in flow-charting decisions are to be taken at specified points. They are shown as Decision Boxes.

While entering the examination hall, sometimes, you have to declare that you do not contain objectionable articles such as notebooks, books etc. In case you produce some script of paper and start scribbling on it as rough work, it may be possible that you are held for cheating or turned out of the hall.

Similarly, some languages require you to declare your belongings (variables, in computer terminology) before proceeding on the programming. You may or may not use them but their declaration is necessary.

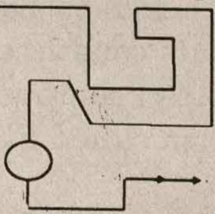
Lastly, the use of arrows. They show the direction of flow of information.

FLOWCHARTS

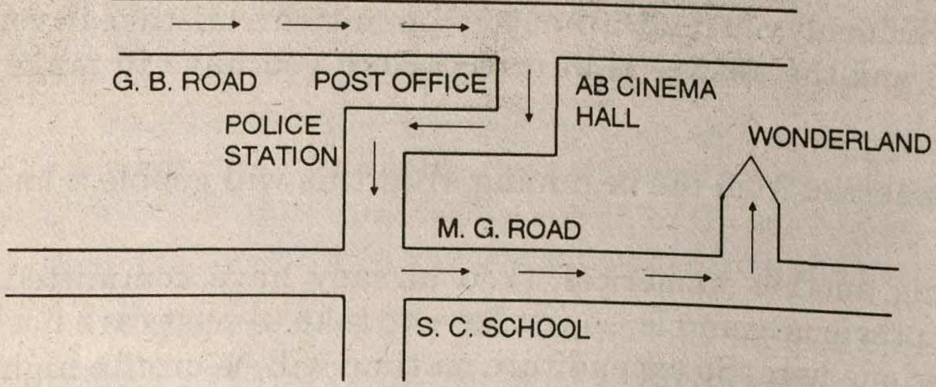
Flowcharts does not mean the charts that flows around. But it is the term used to refer the drawing of figures to solve any problem.

Suppose I wish to send Alice to an unknown place called wonderland and I give her a rough sketch of this kind :-

Then I will take 5% guarantee of her reaching the wonderland.



where as if I give the following map:-



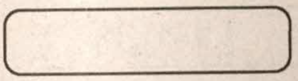
I am 99% assured that she will find her wonderland. Therefore our problem of guiding Alice was solved not only by figures & arrows but labelling or naming them. In computers, the method of solving problems by labelled figures & arrows is called **Flowcharting**.

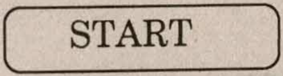
In short, the diagrammatic representation of any problem is flowcharting. This step-by-step guide is used by any programmer when he plans the processing of any question. These are constructed by using special symbols. Each symbol is connected with the other by means of arrows which shows the flow of data or information and are thus called flowlines. These flowlines should always point downwards.

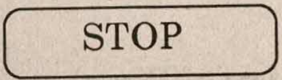
The first formal flowchart was drawn in 1945 by John Von Neuman.

Flowcharting Symbols

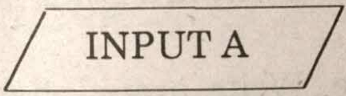
1. **TERMINAL BOX**:- OVAL SHAPE: This symbol can be used to Begin or End the flowchart.

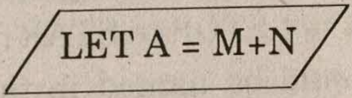


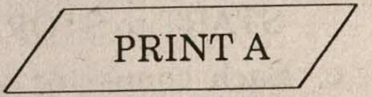




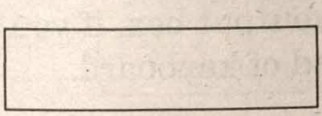
2. **INPUT/OUTPUT BOX** :- PARALLELOGRAM SHAPE. This symbol is used to assign as well as output any information. Thus it represents- LET, INPUT, READ, PRINT etc. statement.

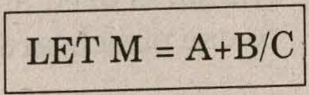


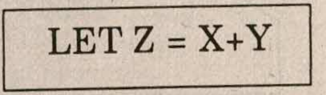




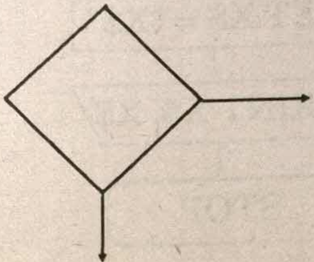
3. **PROCESS BOX** :- RECTANGULAR SHAPE. This symbol indicates various computations/calculations.

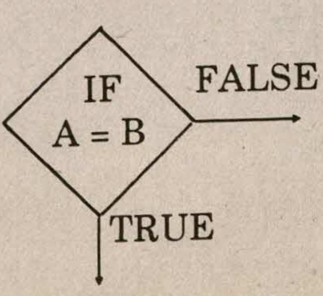


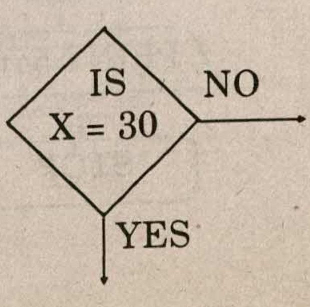




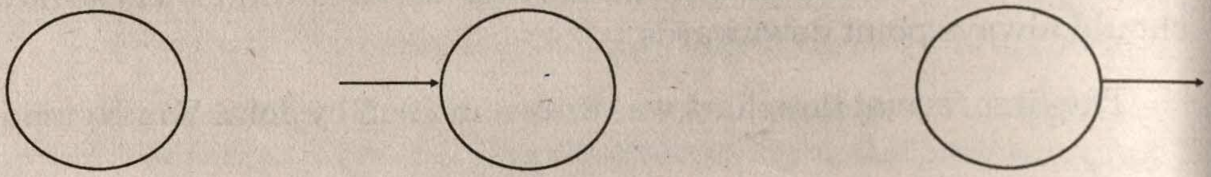
4. **DECISION BOX**:- DIAMOND SHAPE : Whenever any condition is to be checked by the user IF-THEN statement is used in programming and check box is used in flowcharting.



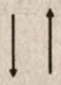




5. **CONNECTORS** :- ROUND SHAPE: To represent GOTO statement connectors are used. It connects the different parts of the flowchart. If an arrows goes in, it is called inconnector & an arrow comming out of it outconnector.



6. **DIRECTIONAL-ARROWS** : This symbol is used to connect different symbols in a flowchart thereby showing the flow of the flowchart.



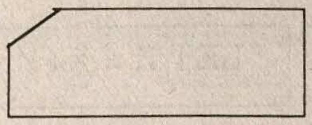
Points(Tips) to remember.

a. Each symbol of the flowchart should be properly connected by arrows with the other symbol and no flying figures.

b. The flowline of any flowchart should always be downwards that is from START to STOP & not STOP to START.

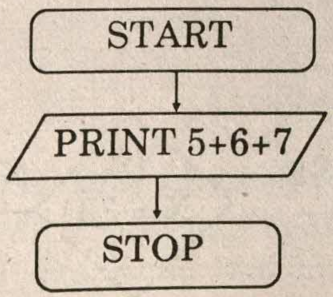
c. Each connector should be named in the circle. Write any alphabet or number in it.

d. It is better to specify the name of the statement in every symbol.

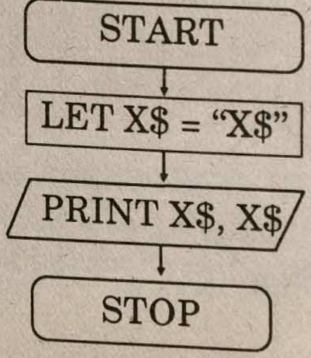
e.  symbol is used as Input/output box, if you wish to use punchcard instead of keyboard.

EXAMPLES

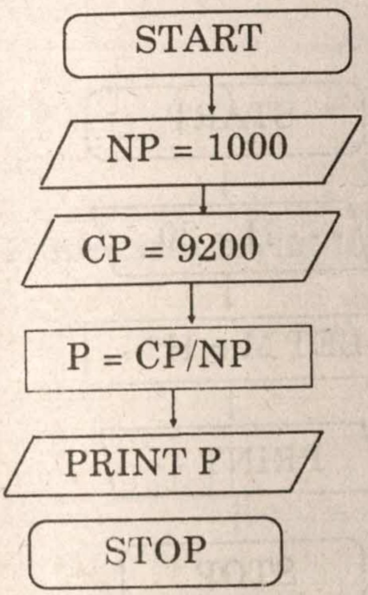
1. Flowchart to add 3 numbers



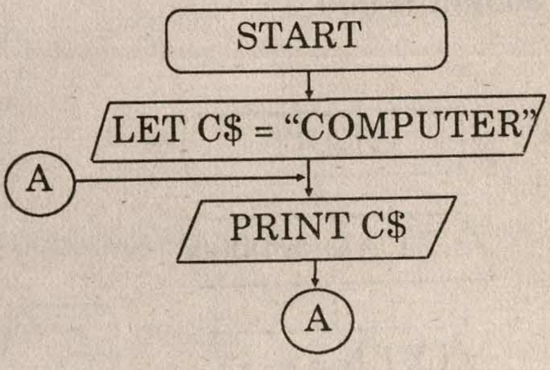
2. To print your name twice.



3. 1000 pens cost Rs. 9200/- F.C. to calculate the price of each pen.



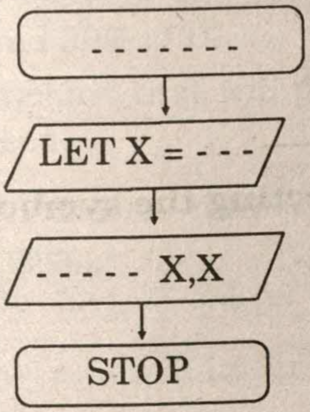
4. Print "computer" word endlessly on the screen.



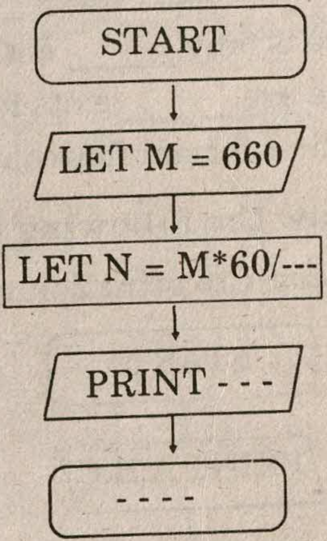
EXERCISE – 5

I. Fill in the blanks:-

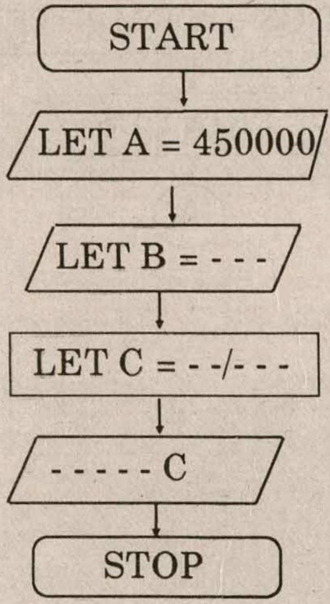
1. F.C.to print the value of X which is 70,000 twice on the screen.



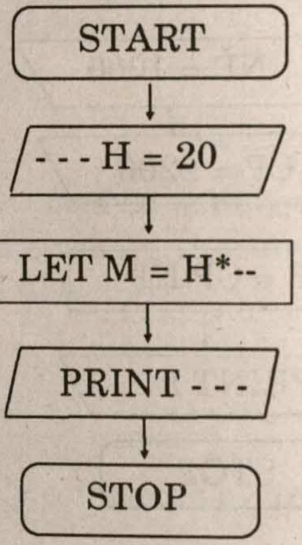
2. F.C. to print 60% of Rs 660.



3. F.C. to calculate the price of each fan when the total cost is 4,50,000 and the number of fans bought is 900.



4. Convert the time 20 hrs into minutes:

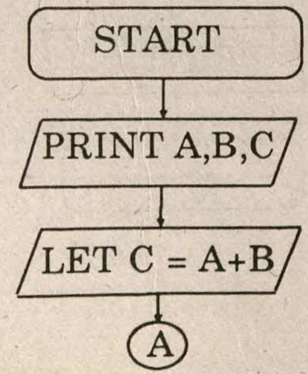


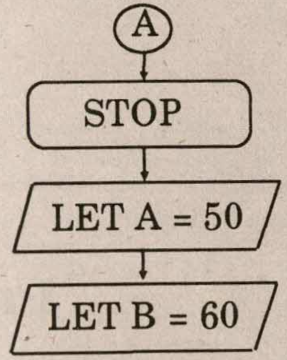
II. Rewrite the following statements by filling in the blanks:-

1. In a flowchart, the flow of data should always be __________.
2. Every flowchart has a _______ symbol in the beginning and at the end.
3. There is only ______ entry into a process box.
4. There are ______ exits for every decision box.
5. The first formal flowchart was drawn by ______.

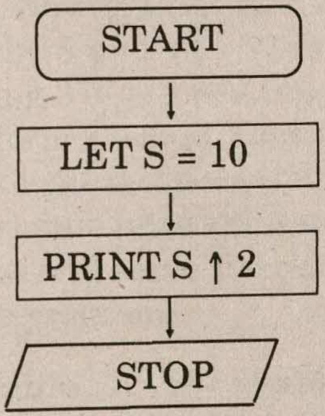
III. Draw the following flowchart by correcting the symbols.

1. Flowchart to print 2 numbers and their sum:-





2. Flowchart to print the area of a 10cms square.



IV

1. Draw a flowchart to find the average of 60,70 &30.
2. The value of A is 325 and B is 635. Draw a flowchart to calculate 20% of A and 30% of B.
3. Flowchart that will print the following result on 2 seperate lines of the screen.

On line 1- total cost of 4 items at Rs. 40/- each & 5 items 50 paise each.

On line 2- 2% of 2.2 .

4. Draw a flowchart for :-

```
10 LET Y = 5
20 PRINT Y*5.5
30 LET Y = 6
40 PRINT Y*60
50 END
```

7

INPUT Statement

Let's play a simple guessing game in which you have to guess any two digit number that is (10 to 99). Now multiply the number by 2 thereby adding 2 to it. Further multiply it by 5 and add 5 to it. Lastly substract 1 from the first 2 digits. The remaining 2 digits was the number you guessed. If I wish to write it in a program form I would have to use a statement which will everytime allow the user to guess different numbers according to his choice and in turn the computer should return the answer that disclose the number guessed. INPUT statement will serve the purpose and allow a direct interaction of the user with the program.

Another example is **Table** : a very familiar term. Most of you must be knowing by heart! An easy way to find the result of 12 multiplied by 7,13 multiplied by 9 and so on. How about writing a program to get these. Okey, here we go, write the one that gives the result of 7 multiplied by 8.

```
10 LET X = 7
20 LET Y = 8
30 PRINT X*Y
40 END
```

But, you see the problem. For every multiplication you have to rewrite some necessary statements everytime. There can be infinite possibilities of multiplication, division & so on. There has to be some way of applying a short cut. Indeed, there is one. The one known as the INPUT statement. How this INPUT works? Well, first of all we define some variables. Remember X,Y,Z are in general known as variables that can assume any value. After defining these variables we write statements where we attach the relationship such as X as the numerator & Y the denominator & Z as the resultant of X divided by Y. Now we introduce the formula and ask the computer to print the result.

```
10 INPUT X,Y
20 LET Z = X/Y
```

```
30 PRINT Z
40 END
```

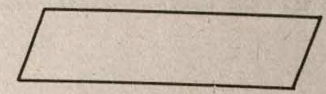
Now the moment you run your program, the computer asks you to give the values of X the numerator and Y the denominator and lo and behold! you get the answer in a jiffy. So simple, so fascinating. No more repeating statements. A program to take care of all problems of division. No doubt that you can device similar ones for multiplication addition and so on.

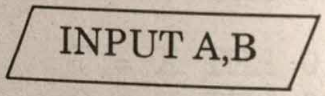
Thus main function of INPUT statement is to assign the values to the variables. The values to the variables are given at the time of program execution. The values thus typed in are assigned to the variables specified in the INPUT statement.

While executing any program having INPUT statement the computer prints a question mark on the screen that is it asks for the value. It waits as long as the information is typed and RETURN (ENTER) key is pressed. It stores the information typed as the value of variable named after the INPUT statement and return the desired answer.

Format(Syntax): 1 INPUT v
where 1 = any linenumber
v = any variable name.

Flowchart symbols





Examples:-

Input statement can be used to assign values to numeric variables :

a. 100 INPUT J
b. 120 INPUT JM

Input statement can be used to assign values to string variable:-

a. 120 INPUT G$
b. 20 INPUT LJ$

When more than one variable is specified in an INPUT statement, the variables and their corresponding information(values) must be seperated by commas.

```
a. INPUT A,B$
b. INPUT M$,N$,C,G
```

It is possible to give a message in the form of a prompt string before the variable name in INPUT statement as it will be printed before the question mark during execution making it clear to the user what is he suppose to type after the question mark.

```
40 INPUT “TYPE ANY NUMBER”;N
70 INPUT “WHAT IS YOUR NAME”;N$
```

Illustrative programs and the output :

1. Program to find the square of any number.

```
10 INPUT “TYPE THE NUMBER”;X
20 PRINT X↑2
30 END

RUN
TYPE THE NUMBER? 4
16
```

2. Program to add any 3 numbors.

```
10 INPUT X,Y,Z
20 PRINT X+Y+Z
30 END
RUN
? 4,2,3
9
```

3. Program to print a welcome message.

```
10 INPUT “ENTER YOUR NAME”; B$
15 CLS
20 PRINT “HELLO!”
30 PRINT “WELCOME”;B$
40 PRINT “NICE MEETING YOU”
50 END
RUN
ENTER YOUR NAME? LENA
(Screen gets cleared)
```

```
HELLO!
WELCOME  LENA
NICE MEETING YOU
```

5. Program to print first 5 multiplies of any number being accepted.

```
10 CLS
20 INPUT “TYPE ANY NUMBER”;A
30 PRINT A,A*2,A*3
40 PRINT A*4, A*5
50 END
RUN
(On a cleared screen)
TYPE ANY NUMBER ?7
7  14  21
28 35
```

EXERCISE – 7

I. Dry Runs :- What will be the output of the following programs?

```
10 REM *any 2 * number*
20 INPUT P
30 INPUT Q
40 PRINT P*Q
50 END
```

2.

```
10 INPUT X,Y
20 LET Z=X/Y
30 LET R=X+X
40 PRINT Z,R
50 END
```

3.

```
10 PRINT “TYPE YOUR NAME”
20 INPUT NA$
30 PRINT NA$, “THE GREAT”
40 END
```

4.

```
10 REM ******
20 INPUT A,B
30 PRINT A+B
40 PRINT A-B
50 PRINT A*B
60 PRINT A^2, B^2
70 END
```

5.

```
10 INPUT “NAME”;A$
20 PRINT “U R 2”; “SWEET”,A$
30 PRINT “SORRY”
40 END
```

6.

```
10 INPUT B
20 PRINT B, B*2, B*3
30 PRINT B*4, B*5, B*6
40 END
```

7.

```
10 REM *FRIENDSHIP*
20 INPUT “MAY I KNOW YOUR GOOD NAME?”; M$
30 PRINT “HI!”, M$
```

```
40 INPUT “HOW ARE YOU”;S$
50 PRINT “HOW NICE YOU HERE”,S$
50 PRINT “NICE TO MEET YOU”, M$
60 PRINT “THANK YOU”: END
```

8.
```
10 INPUT “TYPE ANY NUMBER”N
20 PRINT “SQUARE OF IT IS”,N^2
30 PRINT “5 TIMES OF IT IS”, N*5
40 END
```

9.
```
10 INPUT “RATE AND AMOUNT IN DOLLARS FOR
20 INPUT CONVERSION OF DOLLARS TO RUPEES”,R1,D
30 LET F = R1*D
40 PRINT F
50 END
```

II. DEBUGGING :- Correct the following programs :

1.
```
10  INPUT P=5
20  PRINT P+100
30  END
```

2.
```
10 INPUT R*6
20 END
```

3.
```
10 INPUT A+B
20 PRINT A,B
30 END
```

4.
```
10 INPUT Q
20 Q+Q=R
30 PRINT QR
40 END
```

5.
```
10 INPUT M
20 LET M=6
30 PRINT M/6:M
40 THE END
```

6.
```
10 REM-AREA OF A-
20 REM-RECTANGLE-
30 INPUT “LENGTH”;L
40 INPUT “BREADTH”;B
```

```
50 INPUT A=L+B
60 PRINT AREA
70 END
```

III. Place the correct linenumbers alongside the instruction :--

```
PRINT X,Y
INPUT X,Y
END
```

```
END
PRINT Z
LET Z=L*L
INPUT L
```

```
PRINT "LOVE COMPUTERS"
PRINT "DO YOU"
END
INPUT "TYPE YES OF NO";N$
PRINT "O.K. THANK YOU!"
```

```
REM ENQUIRY
END
PRINT "YOU ARE LUCKY TO BE IN"; N$
INPUT N$
PRINT "YOUR SCHOOL'S NAME PLEASE:"
PRINT "OH! WHAT A GOOD SCHOOL"
```

```
PRINT "SUM OF IT = ";S
LET J=60
END
INPUT K
LETS=J+K
```

Programs :-

1. WAP to find the double of any number.
2. WAP to find the square &cube of any number.
3. WAP to find the total cost of sales according to the price and quantity.

4. WAP that accepts any persons name, age, nationality and address and prints them on seperate lines of the screen.
5. WAP that accepts two numbers J &K. P is the product of J and K. The output should displayed as - - - - - - the two numbers and the value of M which is 10 times of P.

Flowchart

1. Draw a flowchart that accept any 4 numbers and prints their average.
2. Draw a flowchart that converts any time given in hours & minutes into seconds only.
3. Draw a flowchart- The salesman of ABC Co. gets a commission of 10% on the amount of sales. The output should be the name of the salesman followed by the commission earned.

8

GOTO Statement

"Doordarshan". This creature is found almost in every home. You also must be watching it, maybe for your favourite serial, a sports match or even the news. Now-a-days you must3 have noticed that 2 readers deliver the news turn by turn. Sometimes, if a particular news item is to come from some distant kendra, a reference about it is made and suddenly you see another face giving the detail. At the end, the person says "Over to Delhi for the next item". What is this called? In simple terms it is a change of control. For whatever reson, the work is transfered to others. Similarly in programming, it is possible to transfer the flow of execution of statements to a different statement no matter how far it may be situated. May be that you want a particular task to be finished before continuing on the present one.

As explained earlier in chapter 2, you as a person may be commanded over many a times by others. You may have heard & obeyed many of the following commands :-

GOTO SCHOOL
GOTO THE MARKET
GOTO YOUR CLASS
GOTO THE COMPUTER ROOM
GOTO BED ETC.

In the same manner you too can pass commands on the computer. Sometimes you may come across a situation wherein you may like to do a calculation or printing repeatedly for which you have to send the control of the program from one part to another. In such a case, you can use GOTO statement for a basic program. Here you won't be saying GOTO a particular place as the computer can't walk. But you can ask the control of the program while executing to jump from one linenumber to another. These tranfers may involve any checks (Condition as used with IF THEN statement Ch-11) or without any conditions. Therefore GOTO statement is often called UNCONDITIONAL BRANCHING statement.

For example :

40 GOTO 70 {The computer execute the program upto linenumber 40 and from there it skips few lines and starts executing the line from 70 and onwards.

Format : l GOTO n
where l = any linenumber
n = any linenumber other than l.

The space between GO and TO is optional in some computers but not in BBC micro computers and IBM PC.

If the value of n is greater than l then execution is transferred down wards & vice versa.

Examples:-

```
10 GOTO 100
90 GOTO 40
```

Program:-

```
10 PRINT 500
20 GOTO 10
```

```
RUN
500
500
500 (and so on endlessly)
```

To stop this endless pattern either use IF-THEN statement (see Chapter-11) in the program or press ESCAPE key (in BBC computer) or CONTROL and SCROLL LOCK keys (in IBM PC)

```
10 INPUT X,Y
20 PRINT X+Y
30 GOTO 10
40 END
```

If you are expected to visit two or more places at the same time,you will call it impossible. Similarly the control of any program cannot be send to two or more linenumbers at the same time with a single GOTO statement.

Example :-

```
30 GOTO 100, 200, 50
```

You cannot send anyone abruptly to an unknown place which will sound vague. In the same way one cannot sent the control of the program to a linenumber which is not present in the program.

Example :-
INVALID

```
10 PRINT 50+100
20  GOTO 100
```

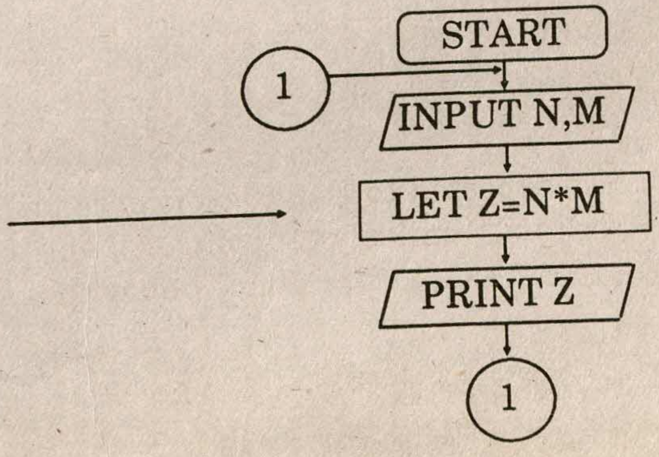
Some more valid example of GOTO statement are as follows:-

1.
```
10 INPUT X
20 PRINT X+X
30 GOTO 10
40 END
        RUN
        ?2
        4
        ?5
        10
```

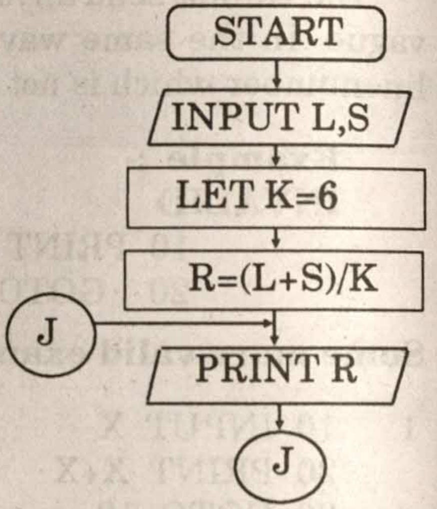
2.
```
10 LET A$="YES"
20 PRINT A$
30 GOTO 20
40 END
```

3.
```
10 CLS
20 PRINT : PRINT "GOTO"
30 LET A=420
40 PRINT A
50 GOTO 10
60 END
```

4.
```
10 INPUT N,M
20 LET Z= N*M
30 PRINT Z
40 GOTO 10
```



5.
```
10 INPUT L,S
20 LET K=6
30 LET R=(L+S)/K
40 PRINT R
50 GOTO 40
```



6.
```
10 REM *SIMPLE INTEREST FOR A GROUP OF PERSONS*
20 INPUT RATE
30 PRINT "NAME, PRINCIPLE & TIME"
40 INPUT X$,PR,T
50 LET SI=PR*RATE *T
60 PRINT X$,SI
70 GOTO 30
80 END
```

EXERCISE – 8

I. Dry Runs: What will be the output of the following programs:-

1.
```
10 INPUT B
20 PRINT B/2
30 GOTO 10
```

2.
```
10 REM ADD ANY 4 NUMBERS
20 INPUT E,F,G,H
30 PRINT E+F+G+H
40 GOTO 20
```

3.
```
100 REM ****
200 PRINT "YOUR NAME"
300 INPUT U$
400 PRINT "WELCOME!"
500 GOTO 300
```

4.
```
10 PRINT 600
20 PRINT 6000
30 GOTO 10
40 END
```

5.
```
10 PRINT 10;0
20 PRINT 0;10
30 GOTO 10
```

6.
```
10 PRINT 1,2,3
20 GOTO 10
30 END
```

7.
```
10 INPUT X
20 INPUT Y
30 PRINT X-Y
40 GOTO 20
```

8.
```
10 LET M=360
20 INPUT D
30 PRINT D+M;"**"
```

```
40 GOTO 20
50 END
```

II. Debugging the following programs:-

1.
```
10 REM* REM*
20 PRINT 6+6
30 GOTO 10+20
```

2.
```
10 REM *JUMP*
20 LET V = 2
30 PRINT "V",V
40 GOTO 200
```

3.
```
10 INPUT X,X
20 PRINT X/Y+X
30 GOTO 30
```

4.
```
10 LET P=3000
20 LET D$ = 20
30 LET K = P+D
40 PRINT K,P
50 GOTO 40,50
```

5.
```
10 INPUT "M"
20 PRINT "M=",M
30 JUMP 20
40 END
```

III. Programs:-

1. WAP to print your name endlessly on the screen.
2. WAP to print the following message flashing on the screen endlessly "THIS WILL NEVER STOP"
3. WAP to print lines of "*",s
4. A shopkeeper allows a discount of 20% of list price on the sale of any item. WAP that calculates the amount of discount and selling price of any item for a group of customers.

VI. Flowcharts

1. Draw a flowchart to print the name of your school endlessly on the screen.
2. Draw a flowchart to print out an endlessly pattern of numbers as follows:-

```
10     20   30
40     50   60
40     50   60
40     50   60 (and so on).
```

9

Functions

Shortcuts are a way of life. Everyone does it. Remember the so- called Donkey-Theorem "Even a donkey knows that it is easy to cross diagonaly than to cross the side a and side b in a triangular path. "The Tables" memorise as many tables as possible, and see how it helps to shorten the time taken to solve numericals.

Similarly, in BASIC some useful and self-explanatory (3 letter) words are provided that simplify procedures and make life simple. These are pre-defined and stored permanently in the memory of that computer and could be recalled as and when required. These functions not only shorten the programs but also makes it easy (simplified) to understand. The only limitation being that these are to be used in conjuction with certain statements, as is given here :- LET/PRINT. Remember, never leave any space between function name and opening bracket.

Some of the functions are listed below:-

(I) **SQR** Its full form is square root. This function calculates the positive square root of any number. Its format is :- SQR (X)

Where X is any number of whose square root is to be found.

Examplory Program

1.
```
10 LET X = 5
20 PRINT SQR(X)
30 END
RUN
2.236068
```

2.
```
100 PRINT SQR(16)
200 END
```

```
RUN
4
```

(II) **INT** Its full form is integer. This function calculates the integral value of any number. Integral value means whole number without any decimal or fractional portion. The number itself is the integral value of that number in case of whole numbers. eg :-

INT(6) = 6 INT(-8) = -8

For positive fractional numbers just hack off the decimal portion. eg:-

INT(+5.7) = 5 INT(9.2999) = 9

In case of negative fractional numbers the computers takes the next lower negative integer. eg:-

INT(-4.2) = -5 INT(-7.73) = -8

Examplory Program

1.
```
10 PRINT INT(-6.6666)
20 END
RUN
-7
```

2.
```
10 REM!REVERSE ANY TWO DIGIT NUMBER!
20 INPUT X
30 Y = X/10
40 Z = INT(Y)
50 A = X-(Z*10)
60 PRINT A;Z
70 END
RUN
? 62
26
```

3.
```
10 CLS
20 LET A = -38.68
30 LET B = -39
40 PRINT B;"IS THE INTEGER OF";A
50 PRINT B-A
60 END
```

```
RUN
-39 IS THE INTEGER OF -38.68
-.32
```

There are various utilities of INT function eg: to device prime number, perfect numbers etc.

(III) **DIV** Its full form is Divide. This function returns the number of times one number gets divisible by another. Division is the same but it does not go to the extent of decimal portion. Remember, if the numbers are not whole numbers, then they are first converted into integers before the division takes place.

Examplory Program

1.
```
10 PRINT 15 DIV
20 END
RUN
3
```

2.
```
100 PRINT 12 DIV 5
200 END
RUN
2
```

3.
```
10 LET X = 12.6
20 LET Y = 7.95
30 PRINT X DIV Y
40 END
RUN
1
```

(Note: Div function works only on BBC micro computer)

IV **MOD** It means remainder. This function returns the remainder. When one number is completely divisible by another then it gives the answer as zero. Remember if the numbers are not whole numbers (integers) then first they are converted into integers and then division takes place to get the remainder.

Examplory Program:-

1.
```
10 PRINT 15 MOD 5
20 END
>RUN
0
```

2.
```
10 PRINT 12 MOD 5
20 END
RUN
2
```

3.
```
10 LET A = 12.6
20 LET B = 7.95
30 PRINT A MOD B
40 END
RUN
5
```

(V) **SGN** Its full form is sign. This function finds whether a number is positive, negative or zero. One can expect any one of the 3 answers (output) while using SGN.

If number is +ve = 1
If number is -ve = -1
If number is Zero = 0

Examplory Program :-

1.
```
10 PRINT SGN(7.897)
20 END
RUN
1
```

B.
```
10 PRINT SGN(-8.2)
20 END
RUN
-1
```

C.
```
5 LET M = 0
10 PRINT SGN(M)
```

```
25 END
RUN
0
```

(VI) **VAL** Its full form is value. This function is used to extract numbers (if any) from a given string. Val function is able to extract numbers only if they are in the beginning of the string. The moment it meets any symbol or letter it stops the extraction. If there are no numbers or if the number is present after a letter or symbol in a string then it returns the answer as zero. But it accepts decimal point, space, positive or negative symbols only. Again the + or – sign should be present in the beginning of the string.

Examplory Program :-

1.
```
10 LET A$ = "30 APPLES"
20 PRINT VAL(A$)
30 END
RUN
30
```

2.
```
100 J$ = "BANANAS = 90"
200 M = VAL(J$)
300 PRINT M
400 END
RUN
0
```

3.
```
2 PRINT VAL ("*2*")
4 END
RUN
0
```

4.
```
10 LET M$ = "3.3CMS"
20 LET L = VAL(M$)
30 PRINT M$,L
40 END
RUN
3.3 CMS      3.3
```

5.

```
10 PRINT VAL("-4.65A")
20 PRINT VAL("+65B")
30 END
RUN
-4.65
+65
```

6.

```
10 LET YOURAGE$ = "30 YEARS"
20 LET MYAGE$ = "25 YEARS"
30 PRINT VAL(YOURAGE$)-VAL(MYAGE$)
40 END
RUN
5
```

(VII) **STR** Its full form is string. This function is used to change a numeric variable into a string variable. It is somewhat opposite of VAL function.

Examplory Program

1.

```
10 LET A = 76
20 LET B$ = STR$(A)
30 PRINT B$
40 END
RUN
76
```

B.

```
10 LET B = 2.2
20 LET D$ = STR$(B)
30 PRINT D$+D$
40 END
RUN
2.22.2
```

(VIII) **ABS** Its full form is absolute. This function is used to find the absolute value of any number. An absolute value of a number is the number itself without any sign.

Examplory Program:-

1.

```
10 LET L = -35.65
20 PRINT ABS(L)
```

```
30 END
RUN
35.65
```

2.
```
100 PRINT ABS(909)
200 END
RUN
909
```

3.
```
100 PRINT ABS(0)
200 PRINT ABS(+27)
300 END
RUN
0
27
```

(IX) **TAB** Its full form is tabulation. The size of every monitor (screen) varies from computer to computer. For example-BBC micro computer has 40 columns and 25 rows. This means the maximum number of characters written from left to right = 39 and from Top to bottom = 24 characters.

Usually a program prints the output from the left most corner of the screen and then sets the following items according to commas and semicolons used in between the list of items in PRINT statement. Thus to have a control on the printing of output on the screen, TAB function is used.

TAB(X)

Where X is the column number from which the printing takes place.

Examples:-

```
20 PRINT TAB(5); 50
```
(This will print 50 at the 5 column of the screen).

```
10 PRINT TAB(12); "GOOD"
```
(This will print GOOD from the 12 column)

You can use as many TAB functions as required with a single PRINT Statement.

Example

1.
```
10 PRINT TAB(2); "ALL;" TAB (7); "THE"; TAB (11); "BEST!"
20 END
RUN
b ALL bb THE b BEST!
(Where b shows spaces)
```

2.
```
10 PRINT TAB(15); 7+5
20 PRINT TAB(15); "GOOD DAY"
RUN
 12
 GOOD DAY
```

3.
```
100 PRINT TAB(4); "*"
200 PRINT TAB(5); "*"
300 PRINT TAB(6); "*"
400 END
RUN
     *
      *
       *
```

If you want to move the output to any specific position on the screen then you not only have to mention the column number but also the row number.

```
TAB(x,y)
Where x stands for column number
      y stands for row number
```

```
10 CLS
20 PRINT TAB(5); "HELLO!"
30 PRINT TAB(10,20); "SPIDERMAN"
40 END
```

```
10 CLS
20 INPUT N$:CLS
30 PRINT TAB(15,5);N$
40 END
```

***Note** : This TAB(X,Y) utility can be used only in BBC micro computer and not in IBMPC. You may use LOCATE Statement instead.

(X) **LOCATE** It is a statement in basic. It can help in printing an item on any position on the screen.

Its format is l LOCATE r,c
Where l can be any line number
r is the row number (1-24)
and c is the column number.

(1-39 or 1-79 depending on the size of the number of character per line.)

The size of screen can be increased (making the characters tiny) or decreased (making the characters huge) with the help of WIDTH statement you can use either 40 width screen or 80.

If you want bold letters then in front of system prompt type WIDTH 80 (Press ENTER)

Examples

```
10 CLS
20 LET T = 30
30 LET V = 40
40 PRINT "T+V="
50 LOCATE 10,5
60 PRINT T+V
70 END
```

{The computer will print on a cleared screen T+V= and then from 10th row 5 column 70}

```
10 INPUT X,Y
20 CLS
30 LOCATE X,Y
40 PRINT "*****"
50 END
```

{The computer will print the 5 stars on a cleared screen on any row or any column depending on the choice of the user as INPUT statement is used}

EXERCISE – 9

I. DRY RUNS:- What will be the output of the following programs:-

1.

```
10 LET P = 36
20 PRINT SQR(P)
30 PRINT SGN(P)
40 END
```

2.

```
100 CLS
200 LET M = 30-36
300 PRINT SGN(M),INT(M)
400 END
```

3.

```
90 PRINT ’TAB(6); “YES”
100 PRINT TAB(6); “****”
110 PRINT TAB (9); “MASTER”
120 PRINT TAB(9); “******”
130 END
```

4.

```
10 INPUT C
20 PRINT INT(C), SGN(C)
40 PRINT TAB(C);“SEE”
50 END
```

5.

```
10 PRINT 13 DIV 3
20 PRINT “ANSWER”
30 END
```

6.

```
10 CLS
20 PRINT TAB(4), INT(-76.9)
30 END
```

7.

```
10 LET L = 70 MOD 8
20 PRINT INT(L)
30 END
```

8.

```
10 INPUT X,Z
20 PRINT INT X, DIV Z
```

```
30 PRINT X MOD Z
40 END
```

II. Programs

(1) Write a program to find the square root and integral value of 8.65

(2) Write a program to print your name on the centre of the screen.

(3) Write a program to find the sign of any number input.

(4) Write a program to find the following on the screen.

```
  *
 ***
______
```

(5) Write a program to print "GEM" on any position of the screen depending on the choice of the user.

(6) The value of D is 98 and E is 89. WAP to print the value of D from the 6th column and E from the 16th column.

(7) Write a program to print the following:-

```
BBBBB
 AAAA
  SSS
   II
    C
```

(8) WAP to find the square root of the integers of the decimal values which the user will input.

III. Flowcharts

1.
```
10 INPUT C1
20 CLS
30 PRINT TAB(C1);C1
40 END
```

2.
```
10 LET G = 200
20 LET H = 300
30 PRINT H DIV G
40 END
```

3.
```
10 PRINT SGN(-8.8)
20 PRINT INT(-8.8)
30 GOTO 10
```

4.
```
10 CLS
20 PRINT TAB(10,12) "**"
30 PRINT "--"
40 GOTO 10
50 END
```

10

Reforming Strings

Till now you had a lot of programming and learnt various new terminologies. How about adding various numbers like 2+2, 23+34 etc. Now you may say, to verify the truth let us write a program to add 23 to 34 and A to B.

```
10 PRINT 23+34
20 END
>RUN
57
```

```
10 PRINT "A" + "B"
20 END
>RUN
AB
```

Now try this program

```
10 PRINT "T" + "V"
20 END
>RUN
TV
```

See you can trick the computer to add the letters T and V to get a combined word "TV". You must have noticed the quotation marks (") that we have put to enclose both T and V letters. Here, you are in for another surprise. Consider the following program:-

```
10 PRINT "2" + "4"
20 END
>RUN
24
```

Amazed? To understand, what happened or what the quotation mark (") actually perform that allow these funny things to happen. You should at this stage know something about strings. The process of adding two or more strings (words) together is known as concatenation.

```
10 LET X$ = "RE"
20 LET Y$ = "CORD"
30 PRINT X$+Y$
>RUN
RECORD
```

```
10 LET M$ = "55"+"5"
20 PRINT M$
30 END
>RUN
555
```

```
10 LET M$ = "50"
20 LET C$ = "CUPS"
30 LET E$ = M$+C$
40 PRINT E$
50 END
>RUN
50CUPS
```

Thus playing up with alphabets (strings) is also possible. It is a great fun. In Basic one can manipulate strings in various manners. The functions, while playing with strings are collectively known as **String Manipulators.** Some of them are listed below:-

1. **LEN Function:-** The full form of LEN is length. This function is used to find the length of any string. In other words, it calculates number of characters present in a string. Any blank spaces if present in any string is also counted. **Format** : LEN(X$)

Where X$ is any string whose length is to be found.

Examples

1.
```
PRINT LEN("YORK")
20 END
>RUN
4
```

2.
```
10 LET X3$ = "COMPUTER=200"
20 LET X = LEN(X3$)
30 PRINT X
40 END
>RUN
12
```

3.
```
100 PRINT LEN("COMPUTER'S BASIC")
120 END
>RUN
16
```

2. **LEFT$ Function:-** This function is used to extract the left most characters of any string.

Format: LEFT$(N$,N)

Where N$ = either a string constant or a string variable
N = number of characters to be extracted (removed) from the Left

If n is greater then the total numbers of characters present in the string then the whole string is printed.

Examples:-

1.
```
10 PRINT LEFT$("MOTHERLAND",6)
20 END
>RUN
MOTHER
```

2.
```
10 LET D$ = "PEACOCK"
20 PRINT LEFT$(D$,3)
30 END
>RUN
PEA
```

3.
```
5 LET M$ = "TIT FOR TAT"
15 LET N$  = LEFT$(M$,5)
25 PRINT N$
35 END
```

```
>RUN
TIT F
```

3. **RIGHT$ Function:-** This function is used to extract characters from the right side of any string.

Format: RIGHT$(N$,N)

Where N$ = either a string constant or a string variable.
N = number of characters to be extracted from the right of the string.

Examples

1.
```
10 PRINT RIGHT$("MOTHERLAND",4)
20 END
>RUN
LAND
```

2.
```
10 LET D$ = "PEACOCK"
20 PRINT RIGHT$(D$,5)
30 END
>RUN
ACOCK
```

3.
```
5 LET K$ = "ALL THE BEST"
15 LET B$ = RIGHT$(K$,8)
20 PRINT B$
25 END
>RUN
THE BEST
```

4. **MID$ Function:-** This function extracts characters from anywhere even from the centre of any string.

Format: MID$(N$,P,M)

Where N$ = any string constant or a string variable
P = Position from which extraction takes place.
M = number of characters from (P) position to be extracted

If M is not specified all the remaining characters of the string from (P) position is extracted.

Examples

1.
```
10 PRINT MID$(“EXAMPLES”,3,5)
20 END
>RUN
AMPLE
```

2.
```
100 LET Q$ = “QUEENSLAND”
200 LET R$ = MID$(Q$,6,7)
300 PRINT N$
400 END
>RUN
SLAND
```

3.
```
5 LET D$ = “8TH FEBRUARY”
15 PRINT MID$(D$,5)
20 END
>RUN
FEBRUARY
```

4.
```
10 LET S$ = “ST. COLUMBA’S”
20 PRINT MID$(S$,1,3)+MID$(S$,12,2)
30 END
>RUN
ST.’S
```

5.
```
10 REM TO PRINT
20 REM INITIALS OF
30 REM ANY NAME
40 CLS:C = 0
50 INPUT N$
60 PRINT LEFT$(N$,1),“.”;
70 LET X = LEN(N$)
80 LET C = C+1
90 IF C = X THEN END
100 IF MID$(N$,C,1) = “b̸” THEN PRINT MID$(N$,C+1,1);“.”.
110 GOTO 80
```

EXERCISE – 10

I. RUNS : What will be the output of the following programs:-

1.
```
10 LET X$ = “JACK AND JILL”
20 PRINT MID$(X$,6,3)
30 END
```

2.
```
100 PRINT LEFT$(“BECAUSE”,2)
200 PRINT RIGHT(“BECAUSE”,5)
300 END
```

3.
```
10 LET F$ = “FIRST”:LET E$ = “AID”
20 LET M$ = F$+E$
30 PRINT M$, LEN(M$)
40 END
```

4.
```
10 LET D$=“BASIC PROGRAM”
20 PRINT LEN(D$)
30 PRINT D$+“MING”
40 END
```

5.
```
10 LET T$ = “NEWS PAPAER”
20 PRINT LEFT$(T$,3)
30 PRINT LEFT$(T$,4)
40 END
```

6.
```
10 LET LT$ = “2 BOR NOT 2B”
20 PRINT LEFT$ (LT$,1)
30 PRINT RIGHT$(LT$,1)
40 PRINT MID$(LT$,1,1)
50 END
```

7.
```
10 INPUT DM$
20 PRINT LEN(DM$)
30 PRINT LEN(DM$)+5
40 END
```

8.
```
10 LET A$ = “6”:LET B$ = “7”
20 PRINT A$+A$+B$
```

```
30 PRINT LEN(B$)+LEN(B$)
40 END
```

9.

```
10 INPUT N$,M$
20 PRINT N$+M$
30 PRINT LEFT$(N$,2)
40 END
```

10.

```
10 LET A$ = "INDIA IS MY COUNTRY"
20 PRINT LEN(A$)
30 PRINT A$;A$
40 GOTO 30
```

II Fill in the blanks:-

1.

```
10 REM * * * *
20 LET A$ = "BLANK"
30 PRINT "LENGTH OF A$=";
40 - - - - - - - - - -
50 - - - - - - - - - -
```

2.

```
REM ** CONCATENATION **
20 LET C$ = "TYPE"
30 LET ---- = "WRITTER"
40 PRINT  ----- D$
50 END
```

3.

```
10 REM *FIRST 4 LETTERS*
20 LET J$ = "JACKFRUIT"
30 - - - - M$ = - - - (J$,4)
40 - - - - M$
50 END
```

4.

```
10 REM * FIRST & LAST CHARACTER*
20 LET Z$ = "COMPUTERS"
30  - - - LEFT$(---,1)
40 PRINT - - -  (Z$,1)
50 END
```

III Programs

1. WAP to calculate the length of any string being input.
2. WAP to form 2 meaning words out of the string "LUCKNOW".
3. WAP to print the first and the last character of any string which is input(accepted).
4. WAP to calculate the lengths of your first name and surname and print their sum.
5. WAP to print all possible combinations of the word "TIME"
6. WAP to make 3 meaningful combinations of the word "LADY FINGER"
7. WAP that accepts any date as MM/DD/YY and print only the (DD) as the output.

IV Flowcharts

1. Draw a flowchart to extract "RISE" from the word "ENTERPRISES".
2. Draw a flowchart that accepts words "GOOD","NIGHT" and "MORNING" only once and prints GOOD MORNING and GOOD NIGHT.

11

IF-THEN statement

In everyday life, man need to take several decisions in order to be successful in life. Most of the time he keeps a check on his actions by applying some conditions. **For example**

If I work then I will get best division.

If you go to the market then you can buy some fruits.

If I win the game then I will treat you else you will have to give me a treat.

Similarly, at times it becomes very essential to make a check or take up a decision while writing a program for a problem. IF THEN - statement in BASIC serves the purpose of making decisions. This statement helps in transferring the control of the program while executing from one part to another depending on some condition. In a way, it controls the execution of the program therefore it is a control statement.

Format (syntax) of If-then statement is
l IF condition THEN m
Where l = any line number
condition = any relational expression
m = any line number other then l or any statement.

When a program containing If then statement is executed, the condition given in it is tested. Either the condition will be true or false. If the check comes to be true the computer will execute according to what is given after THEN whereas if it comes out to be false what is given on the very next line is executed. The direction of execution of any program depends on the condition.

```
1 IF condition THEN m
 |  ----------------→
 ↓ NO            YES
 (FALSE)        (TRUE)
```

Examples

```
100 IF A = 10 THEN 60
160 IF N > M THEN PRINT N * M
32  IF T$ = "GOOD" THEN LET K = J+J
```

From the above examples it must be clear that every condition uses constants, variables or expressions along with any of the following relational operators >, <, =, <>, >=, <=.

Invalid Examples

```
20 IF A$ = 6 THEN 60
40 IF B <> D THEN 40
80 IF "I AM" > "YOU" THEN "YOU"
10 IF K=B THAN K
28 IF M THEN 210
```

Examplory Programs

1. A program to accept a number and print its square only if it is greater than 30.

```
10 INPUT N
20 IF N > 30 THEN PRINT N↑2
30 END
```

2. A program that accepts the side of any square and prints its area or perimeter depending on the choice of the user.

```
10 INPUT "SIDE OF THE SQUARE";S
20 INPUT "CHOICE : AREA = A, PERIMETER = P";C$
30 IF C$ = "A" THEN 50
40 PRINT "PERIMETER="; S*4:END
50 PRINT "AREA="; S↑2 : END
```

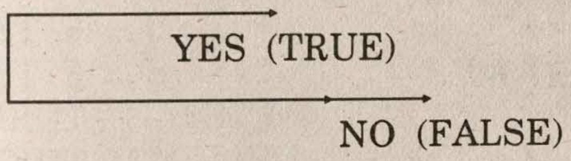
3. A program that accepts the names of a group of persons and prints only those names which starts with "M".

```
10 INPUT “TYPE IN YOUR NAME”;N$
20 R$ = LEFT$(N$,1)
30 IF R$ = “M” THEN PRINT N$
40 GOTO 10
```

IF-THEN-ELSE Statement

The word “ELSE” can be used with IF-THEN statement as a variation or an extension. While executing IF-THEN-ELSE Statement in any program, if the condition is true the control of the program moves to THEN part and if it is false it moves directly to the ELSE part instead of first going to the following line.

```
1 IF condition THEN m ELSE p
```



Examples:-

```
10 IF G = 100  THEN 100 ELSE 250
50 IF B$ = “BASIC” THEN PRINT B$ ELSE PRINT "COMPUTERS"
70 IF C = D THEN PRINT C ELSE END
```

Examplory Program:-

1. A Program to find the smallest number out of any two numbers:-

```
100 INPUT T,S
110 IF T < S THEN PRINT T ELSE PRINT S
120 END
```

2. A program to accept two numbers and print their sum if they are equal else print their difference.

```
10 INPUT A,B
20 IF A = B THEN PRINT A+B ELSE PRINT A-B
30 END
```

Counter System

Let us take an instance where Master lazybee came to school without doing his home work. Miss strictbee, his teacher, punished him by asking him to write the same work 100 times in his note book. He wrote the answers and then counted and numbered them. In this way, his problem was solved. Similarly in computers we need some quantity not like lazybee but some character which can do counting for us. This is possible by using numeric variables in combination with IF-THEN and GOTO statement which will count the number of times a particular process is undertaken. This counter system will be clear from the following example which generates numbers from 1 to 10.

```
10 LET C = 0
20 LET C(1) = C + 1(2)
30 PRINT C
40 IF C = 10(3) THEN 60
50 GOTO 20
60 END
```

Where (1) Shows the first value.
(2) Value by which C increases (Class interval)
(3) Final value.

In the above example, C is working as a counter and at every step the value of C increases by class intervals and gradually stops at the last limit which is 10.

Similarly, If you are being asked to print the word "EXAMINATION" 5000 times on the screen, you may adopt the following long methods and take 3 to 4 hours just to type, print, set and count. Some of you may even get examination fever.

```
10 LET X$ = "EXAMINATION"
20 PRINT X$
30 PRINT X$ [and so on 5000 times]
```

The same problem can be solved by counter method and will hardly take 3 to 4 minutes.

```
10 LET L = 0
20 LET L = L + 1
```

```
30 PRINT "EXAMINATION"
40 IF L = 5000 THE END
50 GOTO 20
```

Examples :-

1. Program to print the multiples of 4 till 200.

```
10 LET M = 0
20 LET M = M + 4
30 PRINT M
40 IF M = 200  THEN END ELSE 20
```

2. A Program to find the total and average marks of 50 students of a class in 3 different subjects.

```
10 LET S = 0
20 LET S = S + 1:LET TT = 0:LET AV = 0: CLS
30 INPUT "NAME";N$
40 INPUT "ENGLISH MARKS";EM
50 INPUT "HINDI MARKS"; HM
60 INPUT "MARKS IN MATHS"; MM
70 LET TT = EM + HM + MM
80 LET AV = TT/3
90 PRINT "NAME = ";N$
100 PRINT "TOTAL MARKS";TT
110 PRINT "AVERAGE MARKS="; AV
120 IF S = 50 THEN 140
130 GOTO 20
.140 PRINT *** THE END *** : END
```

After feeding in the above program while executing quickly just type out the name and marks of each candidate and press ENTER (Return) key after every entry, subsequently the computer will return the total and average marks of each candidate thus putting the teacher and the student to an ease.

AND, OR (Logical Operations)

Sometimes you may need to use more then one IF-THEN statement in a program depending on the number of conditions the program has :-

Example :- Admission to a computer institute "MASTER OF COMPUTERS INSTITUTE, MANGALORE" is given on the following conditions :-

a. The Person should be graduate.

b. Above 20 years of age.

c. Should secure 75% or above in entrance exams

A program for the above problem using IF-THEN statement:-

```
10 CLS
20 PRINT “MASTER OF COMPUTERS INSTITUTE”
30 PRINT “**********************************”
40 INPUT “CANDIDATE’S NAME”; N$
50 INPUT “GRADUATE OR NOT,TYPE Y/N”; G$
60 INPUT “AGE IN FIGURES”;A
70 INPUT “% OF MARKS IN SELECTION TEST”; P
80 IF G$ = “Y” THEN 100
90 PRINT “SORRY”: GOTO 10
100 IF A > 20 THEN 120
110 GOTO 90
120 IF P >= 75 THEN 140
130 GOTO 90
140 PRINT “CONGRATS! YOU ARE ADMITTED” : GOTO 10
```

The above program can be shorten by using logical operators.

```
10 CLS
20 PRINT “MASTER OF COMPUTERS INSTITUTE”
30 PRINT “*********************************"
40 INPUT “CANDIDATE’S NAME”:N$
50 INPUT “GRADUATE OR NOT, TYPE Y/N”;G$
60 INPUT “AGE IN FIGURES”;A
70 INPUT “% MARKS IN SELECTION TEST”;P
80 IF G$ = “Y” AND A > 20 AND P >= 70 THEN 90 ELSE 100
90 PRINT “ADMISSION GRANTED”: GOTO 10
100 PRINT “SORRY !”: GOTO 10
```

Let us now suppose the admission to MCI is relaxed and if a candidate at least fulfills any of the 3 conditions, the admissions are granted. In this case OR logical operator is used.

Program from 10 to 70 line number remains the same. After it type the following

```
80 IF G$ = “Y” OR A > 20 OR P >= 75
   THEN PRINT “ADMISSION GRANTED”: GOTO 10
90 PRINT “SORRY !” : GOTO 10
```

This makes it clear that in case of AND operator the operation is true if all the conditions are individually true. And in case of OR operator the operation is considered to be true if any of the conditions are true. Even if more then one or all are true still the operation is true.

It is clear that

1. In case of AND operator, the operation is true if all the conditions are individually true.
2. In case of OR operator, the operation is considered to be true if any of the conditions are true or even if more then one or all the conditions are true.

Thus IF-THEN-ELSE (AND & OR) statement can solve many problems in a short way.

EXERCISE - 11

I. Dry Runs

1.

```
10 INPUT “TYPE A NUMBER”;N
20 IF N > 10 THEN END
30 PRINT “NUMBER”;N;“IS LESS THAN 10”
40 END
```

2.

```
10 CLS
20 PRINT “CHOICE OF DRINK”
30 IF C$ = “HOT OR COLD, H/C”;C$
40 IF C$ = “H” THEN 60
50 PRINT “YOU CAN HAVE LIMCA, MAAZA, FUNDAY...”:END
60 PRINT “YOU CAN HAVE TEA, COFFEE...”: END
```

3.

```
10 CLS
20 LET S = 5
30 PRINT S
40 IF S = 50 THEN END
50 S = S+5 : GOTO 30
```

4.

```
100 INPUT “ANY 2 NUMBERS”; M,N
200 IF MN THEN 400
300 PRINT N, “IS GREATER”:END
400 PRINT M,“"IS GREATER”:END
```

5.

```
10 CLS
20 INPUT “GIVE ANY NAME”;A$
30 LET G = G + 1 : PRINT TAB(G); A$
40 IF G = 20 THEN END
50 GOTO 30
```

6.

```
10 CLS: PRINT “WHAT IS THE”
20 PRINT “CAPITAL OF ITALY”
30 INPUT I$
40 IF I$ = “ROME” THEN PRINT “WELLDONE”:END
50 PRINT “SORRY! TRY AGAIN”: GOTO 10
```

7.

```
10 REM*EVEN
20 REM*NUMBERS*
30 X = 0
40 X = X + 2
50 IF X = 50 THEN END
60 PRINT X: GOTO 40
```

8.

```
10 CLS
20 PRINT “------------”
30 INPUT “DO YOU WISH TO CONTINUE Y/N”;
40 IF X$ = “Y” THEN 20
50 END
```

9.

```
10 INPUT “ANY NAME”; N$
20 INPUT “HOW MANY LETTERS DOES IT CONTAIN”;N
30 IF N = LEN(N$) THEN PRINT “RIGHT“ ELSE PRINT “WRONG”
40 END
```

10.

```
5 REM-AREA OF A SQUARE-
10 INPUT “SIDE OF SQUARE”; S
15 PRINT S^2; “IS THE AREA”;
20 INPUT “WISH TO CONTINUE Y/N”;W$
25 IF W$ = “Y” THEN 10
30 END
```

II DEBUGGING: Rectify the following programs:

1.

```
10  TO CHECK WHETHER 2 NUMBER’S ARE=
20 LET 0 = 10; LET P = 20
30 IF O IS EQUAL TO P THEN PRINT "YES"
40 PRINT "NO"END
```

2.

```
10 INPUT Q$
20 IF LEN (Q$)10 THAN PRINT “TOO LONG WORD”
30 ENDED
```

3.

```
100 PRINT “*************************”,“GO”
20 LET T = 5
200 T = T + 1
300 IF T = 10 END ELSE 20
```

4.
```
10 REMARK NATURAL NUMBERS 100-200
20 KD = 0
30 KD =  KD + 2
40 PRINT KD
50 IF KD = 200 THEN 30
60 END
```

III. State whether the following statements true/false:

1. 90 IF EF THEN 9000
2. 1000 IF "YES" = Y$ THEN 360
3. 140 IF TTITHEN 140
4. 100 IF T = "TRUE" THEN 170
5. 50 IS FB = BF THEN 22
6. 260 IF L = 25 THEN 261
7. 66 IF LM THEN 88 ELSE 98
8. 300 IF J$ = K$ THEN LET M$ = "MAY"
9. 10 IF GH THEN I
10. 70 IF A+B = 70 THEN 700

IV. Rewrite the following programs by filling in the blanks.

1.
```
10 REM CHECK THE LENGTHS
20 -------- OF 2 STRINGS R = ------
30 INPUT T$ : ----U$
40 IF --------------
50 PRINT "NOT EQUAL": END
60 PRINT "EQUAL":END
```

2.
```
10 REM PRINT NO'S IN DESENDING ORDER 10 to 0
20 LET LL = ---------
30 LET LL = LL ----------1
40 PRINT -------------
50 -----------------
60 GOTO 30
```

3.
```
10 REM PRINT YOUR NAME 100 TIMES ON THE SCREEN
20 W = ----------
30 W = -----------
40 PRINT --------------
50 ----------------
```

```
60 END
```

4.

```
10 REM WORDS MORE THEN 5 LETTERS TO BE
20 REM PRINTED ON CLEARED SCREEN
30 ----------------------
40 INPUT "TYPE ANY WORD";
50 LET QM = -----(Q$)
60 --------------
70 GOTO ------------
```

5.

```
10 REM CHECK FOR MINOR OR MAJOR
20 --------"AGE";A

30 IF ----- = 18 THEN PRINT "MAJOR" -------- PRINT "MINOR"
40 END
```

V. Programs

1. WAP that accepts any set of numbers and prints only those above 100.
2. WAP that accepts the names and age of a set of persons and prints the name of only those persons who are major that is 18 years and above.
3. WAP to generate odd numbers upto 30.
4. WAP to find whether a candidate has passed or failed in an examination. The program should accept the names and marks in 4 different subjects. IF the total marks is 250 or more then he is passed else failed.
5. WAP to print the following series backwards 100, 90, 80, 70 10
6. WAP to print the product of any 2 numbers if their sum is greater then 50 else print the difference.
7. WAP that accepts any word and prints it as many times as required.
8. WAP to find whether any number is positive, negative or zero.
9. WAP to find whether a number being typed by the user is odd or even.
10. Write a guessing game program in which the user is expected to guess the number 39 out of numbers from 1 to 50.
11. WAP using logical operator for the following problem :

The program should accept the name, age and nationality of any person and find whether he is eligible to vote or not when the condition is, if a person is an Indian and is 18 years or older he can vote.

12. WAP that accepts the measurement of any angle and prints whether it is a figure of an acute, right or obtuse angle.

VI. Flowcharts

1. Draw a flowchart to print the square and cube of any number if the number is positive else continue.
2. Draw a flowchart to find greater number out of any 3 numbers.
3. Draw a flowchart that accepts there sides of an triangle and finds whether it is an equilateral, isosceles or scalene triangle.

12

FOR - NEXT Statement

FOR-NEXT statement is a very powerful statement in BASIC. Every FOR-NEXT statement creates a loop. Loop means a certain process (number of steps) which is to be repeated in a program. Thus every FOR-NEXT statement directs the computer to repeat a process for a given number of times. It serves the purpose of counters, GOTO and IF-THEN statement.

FOR-NEXT statement is a combination of two dependent statements FOR and NEXT. You can never have a FOR statement without its NEXT statement or vice-versa. FOR statement is written at the beginning of the loop and NEXT statement is written at the end of the entire loop.

The function of FOR statement is used to assign values to the variables and NEXT statement is used to change (increase or decrease) the value of the variable. FOR statement combines the initialization, incrementation and checking for the final limit of the counters.

The format of FOR-NEXT statement is :-

l FOR k = k TO k2 STEP k3

Where l = any line number
To and STEP = part of FOR statement
k = any variable name
k1 = initial value of k
k2 = final value of k
k3 = incremental value of [class-interval] of k

l NEXT k

Where l = any line number
k = Same variable name as defined in FOR statement

Examples

```
90 FOR L = 2 TO 90 STEP 4
150 FOR M = 190 TO 280 STEP 3
10 FOR R = 10 TO 100 STEP 10
```

Whenever a FOR-NEXT statement is encountered in a program, a value is first assigned to the variable and then the statements (if any) following the FOR statements are executed, till it meets the NEXT statement. As soon as the computer reads NEXT it goes back all over again to the FOR statement, and the value of the variable changes, according to incremental value (class-interval) mentioned after STEP thereby checking the last limit. Again the same process is repeated. As soon as the value of the variable exceeds the maximum limit, the loop stops and the computer starts executing the lines following the NEXT statement.

There are certain rules which are to be kept in mind while using FOR-NEXT statement.

1. The (variable name) whose value is controlled by the FOR-NEXT statement is called control variable. This control variable should always be numeric and not string.
2. Each FOR-NEXT statement will deal with only one variable name at a time & not more.

INVALID

```
10 FOR I, J = 10 TO 100 STEP 2
100 NEXT R,F
```

3. If the value of k3 (incremental value) is 1 that is if the variable name has to be increased by 1 then to mention "STEP 1" is optional. In case it is not mentioned the computer takes it for granted as 1.

VALID

```
200 FOR I = 1 TO 100 STEP 1
70 FOR M = 4 TO 70
```

4. If the value of k2 (final value) is greater then k1 (first value), then k3 has to be positive as it has to increase from k1, to reach k2.

VALID

```
20 FOR G = 6 TO 56 STEP 5
10 FOR L1 = 2 TO 50 STEP 2
```

5. If the value of k2 (final value) is lesser then k1, then k3 has to be negative value as it has to decrease (come down) from k2, to reach k1.

VALID

```
77 FOR XX = 90 TO 9 STEP -1
600 FOR Z2 = 66 TO 22 STEP -6
```

6. k1, k2 and k3 can either be numeric constants, numeric Variables or expressions.

VALID

```
100 FOR MJ = K TO F STEP 2
80 FOR LD = X+2 TO Y STEP M*6
```

7. k1, or k2 can either be positive, negative or zero.

VALID

```
25 FOR DB = 98 TO -98 STEP -8
62 FOR I = 0 TO 5 STEP 2
```

8. If only one FOR-NEXT is used in a program the variable name after NEXT is optional.

VALID

```
10 FOR B = 2 TO 40 STEP 2
20 PRINT B+6
30 NEXT
```

9. The occurrence of one FOR-NEXT statement within another in a program is called nested FOR-NEXT statement (or nested loops program). There can be any number of loops within a loop. Each FOR-NEXT statement should define a different variable name. The only important thing to be kept in mind is that no two loops should cross or intersect. This means that the NEXT for the first FOR statement should come after the NEXT for the second FOR statement.An outer loop and a nested (inner) loop cannot have the same running variable.

The control can be transferred from an inner (nested) loop to a statement in an outer loop or to a statement outside the entire nest. But control cannot be transferred to a statement within a nest from a point outside the nest.

Example Programs

1. A program to print 50 stars on the screen.

```
10 FOR I = 1 TO 50
20 PRINT "*"
30 NEXT I
40 END
```

2. A program to print natural numbers from 1 to 1000.

```
10 FOR RN = 1 TO 1000
20 PRINT RN
30 NEXT RN
40 END
```

3. A program to print the sum of first 10 natural numbers.

```
10 LET SUM = 0
20 FOR NI = 1 TO 10
30 LET SUM = SUM + N1
40 NEXT NI
50 PRINT SUM : END
```

4. A Program to print odd numbers between 11 and 77 in reverse order.

```
100 FOR MUM = 77 TO 11 STEP -2
200 PRINT MUM
300 NEXT MUM
400 END
```

5. A program that calculates the monthly electricity bill for 12 months and prints the total bill for one year for 100 persons.

```
10 FOR A = 1 TO 100
15 LET SUM = 0
20 INPUT "NAME";N$
25 FOR B = 1 TO 12
30 INPUT "RATE PER UNIT";RT
35 INPUT "UNITS CONSUMED FOR THE MONTH";U
40 BILL = RT*U
45 SUM = SUM+BILL
50 NEXT B
55 PRINT N$, SUM
60 NEXT A
70 END
```

EXERCISE – 12

I **Dry Runs :-** What will be the output of the following programs:-

1.

```
10 FOR C = 1 TO 6
20 PRINT C,C
30 NEXT C
40 END
```

2.

```
10 CLS:PRINT “*********”
20 FOR XY = 5 TO 25 STEP 5
30 PRINT “***********”
40 NEXT XY
50 END
```

3.

```
100 PRINT “%”
120 FOR I = 1 TO 6
130 PRINT TAB(I);“%”
140 NEXT I
150 PRINT “THE END”
160 END
```

4.

```
10 FOR FIX = 1 TO 5
20 PRINT “THE NUMBER = ”;FIX
30 PRINT
40 PRINT “ITS SQUARE=”; FIX ↑ 2
50 PRINT
60 NEXT FIX
70 END
```

5.

```
10 FOR M = 1 TO 3
20 FOR N = 3 TO 1 STEP -1
30 PRINT “HELLO”
40 NEXT N
50 NEXT M
60 END
```

6.

```
10 FOR Q = 60 TO 40 STEP -5
20 PRINT Q;Q
30 NEXT Q
40 END
```

7.

```
10 PRINT “CELCIUS”, “FAHRENHEIT”
20 FOR C = 0 TO 10
30 F = (9/5)*C+32
40 PRINT C,F
50 NEXT C
60 END
```

8.

```
10 FOR E = 1 TO 3
20 FOR F = 1 TO 2
30 EZF = (E**2+F**2)/(E*F)
40 PRINT E,F,EZF
50 NEXT F
60 NEXT E: END
```

II. Fill in the blanks:-

1.

```
10 FOR ------ = 6 TO 70---------20
20 PRINT T
30 NEXT -----------
40 ----------------
```

2.

```
50 REM* TO PRINT NUMBERS FROM 100 TO 200
100 FOR Q = ------- TO ------- STEP -------
150 --------- Q
160 --------- Q
170 ---------
```

3.

```
10 REM * PROGRAM TO PRINT 5 STARS IN A LINE
20 FOR ----=------------
30 PRINT “*”;
40 ---------
50 ---------
```

4.

```
10 TO PRINT ANY WORD INPUT 90 TIMES ON THE SCREEN
20 ---------------------
30 FOR -------------------
40 PRINT T$
50 ------- WORD
60 END
```

5.

```
10 REM * PRINT EVEN NUMBERS 80 TO 800
20 REM * IN REVERSE ORDER
```

```
30 FOR D = --------------
40 ----------------
50 NEXT -----------
60 END
```

III. Debugging

1.

```
10 FOR D = 5 TO 50 STEP -5
20 NEXT D
3 END
```

2.

```
10 FOR GD = 50 TO 5
20 PRIN GD
30 END
```

3.

```
100 FOR N = 0 TO -10 STEP -5
110 PRINT M+M
120 NEXT M
130 END
```

4.

```
5  FOR N$ = 2 TO 22 STEP 2
10 PRINT N$,N
15 NEXT N$
20 ENDED
```

5.

```
10 INPUT K
20 FOR J = L TO M STEP L
30 PRINT **
40 NEXT J
50 END
```

6.

```
10 FOR KT + TK = -6 TO 6 STEP-2
20 PRINT KT
30 NEXT TK
40 END
```

7.

```
100 FOR H = 3 TO 33 STEP 3
120 PRINT H
130 NEXT H1
140 END
```

8.

```
10 FOR Z5 = Z5 TO 100 STEP 10
20 PRINT Z5 * 30
```

```
30 PRINT
40 PRINT Z5  2
50 NEXT
60 END
```

9.

```
10 FOR P = 2 TO 20 STEP 2
20 FOR Q = 20 TO 2 STEP 2
30 PRINT P,Q
40 PRINT NESTED LOOPS
50 NEXT P
60 NEXT Q
70 END
```

10.

```
10 K = 0
20 REM SUM OF THE CUBES OF FIRST FIVE
25 NATURAL NUMBERS
30 FOR I = 1 TO 5
40 K = K*K
50 NEXT K
60 PRINT K
70 END
```

IV. Programs

1. WAP to print 1000 natural numbers.
2. WAP to print first 10 even numbers.
3. WAP to print odd numbers upto 40.
4. WAP to print any person's name and age 100 times on the screen
5. WAP to accept any word and print it as many times depending on the choice of the user.
6. WAP to print first twenty multiples of 5.
7. WAP to print first ten multiples of any number input by the user.
8. WAP to print the sum of the squares of first ten multiplies of 3.
9. WAP to print the average of any five numbers.
10. WAP to print the sum of the first hundred natural numbers.
11. WAP that accepts any 50 numbers and prints the total number of numbers greater then 20.
12. WAP to print the first five multiples of 8 in reverse order.

13. WAP that prints out first 6 multiples of the first 6 natural numbers.
14. WAP that accepts any 4 numbers and prints its square root, square and cube of each number.
15. WAP that inputs the price of each item and number of items of each product. It should calculate and print the total cost of 10 products at the end.
16. WAP that accepts any name and prints it in reverse order.

13

FUN Programs

BBC Microcomputer

1.
```
10 REM *TRIANGLE*
20 MODE 4
30 DRAW 600,0:DRAW 0,600
40 DRAW 0,0:END
```

2.
```
10 REM *SQUARE*
20 MODE 5:DRAW 200,0
30 DRAW 200,200:DRAW 0,200
40 DRAW 0,0:END
```

3.
```
10 REM *RAGAS*
20 CLS:FOR P = 1 TO 8
30 READ B
40 SOUND 1,-10,B,5
50 NEXT P
60 DATA 53,61,69,73,81,89,97,101
70 RESTORE:GOTO 20
```

4.
```
10 REM *ALARM*
20 FOR X = 10 TO 100 STEP 10
30 SOUND 2,-15,200,2:SOUND 2,0,0,2
40 SOUND 2,-15,200,2
50 SOUND 2,0,0,9:NEXT X:END
```

5.
```
10 MODE 5
20 CLS
30 FOR I = 1 TO 45
40 LET X = INT(RND(I)*30+I)
50 LET Y = INT(RND(I)*30+I)
60 PLOT X,Y
```

```
65 NEXT I
70 GOTO 20
80 END
```

6.
```
10 INPUT "YOUR NAME";N$
15 CLS
20 PRINT TAB(18,10);"WELCOME",N$
30 GOTO 15
```

7.
```
10 MODE 0
20 MOVE 440,612:DRAW 840,612
30 DRAW 840,412:DRAW 440,412
40 DRAW 440,612:GOTO 20
```

8.
```
10 INPUT "TYPE ANY NAME";T$
20 CLS
30 PRINT CHR$(141);CHR$(136);T$
40 PRINT CHR$(141);CHR$(136);T$
50 END
```

9.
```
10 CLS
20 PRINT TAB(18,12);CHR$(141);"I THE GREAT"
30 PRINT TAB(18,12);CHR$(141);"I THE GREAT"
40 GOTO 10
```

10.
```
10 LET B = RND(1000)
20 MODE 1
30 VDU 23;8202;0;0;0;
40 CLS
50 PRINT B
60 FOR A = 0 TO 1000 STEP 5
70 B = 1000 - B
80 DRAW A,B
90 DRAW B,A
100 NEXT A
110 MOVE 0,0
120 DRAW 0,1000
130 DRAW 1000,1000
140 DRAW 1000,0
150 DRAW 0,0
160 PRINT TAB(0,0)
170 PRINT "PRESS ANY KEY"
```

```
180 A$ = "GET$ TO RUN"
190 GOTO 10
```

IBMPC

1.

```
10 CLS:KEY OFF:SCREEN 2
20 FOR X = 1 TO 640 STEP 1
30 LINE(X,0)-(640-X,200)
40 NEXT X
50 FOR Y = 1 TO 200 STEP 2
60 LINE (1,(200-Y))-(640,Y)
70 NEXT Y
80 LOCATE 13,38:PRINT "WELCOME"
100 END
```

2.

```
10 REM *HALF CIRCLES*
20 SCREEN 1:FOR A = 1 TO 160 STEP 3
30 CIRCLE (160,200),1+A
40 NEXT A
```

3.

```
10 REM *RANDOM MUSIC
20 FOR A = 1 TO 500
30 B = INT(1000*RND+500)
40 SOUND B,-1
50 NEXT:GOTO 20
```

4.

```
10 REM *TO KNOW WHETHER A*
15 REM NUMBER IS DIVISIBLE BY ANOTHER*
20 INPUT "PLEASE TYPE 2 NUMBERS";X,Y
30 LET D = X/Y:C = INT(D)
40 IF C = D THEN 60
50 PRINT X; "IS NOT DIVISIBLE BY";Y:END
60 PRINT X; "IS DIVISIBLE BY";Y:END
```

5.

```
10 CLS:SCREEN 1
20 FOR B = 1 TO 1000
30 PSET(300*RND,190*RND), 2*RND+1
40 LOCATE 10, 15:PRINT "STARS IN THE SKY"
50 NEXT:END
```

6.

```
10 CLS:KEY OFF:SCREEN 2
```

```
20 CIRCLE(320,100),40
30 LOCATE 13,38:PRINT “HELLO!”
40 LINE(280,60)-(360,140),,B
50 GOTO 10
```

7.

```
 5 CLS
10 FOR I = 1 TO 160 STEP 5
20 CIRCLE(0,0),I:CIRCLE(320,0),I
30 CIRCLE(0,200),I:CIRCLE(320,200),I
40 NEXT I
50 LOCATE 14,22 : PRINT “OH!”
60 END
```

Common

1.

```
10 REM *JUMPING MAN*
20 FOR J = 1 TO 25 STEP 2
30 CLS:PRINT
40 PRINT TAB(J);“–O–”
50 PRINT TAB(J);“<  >”
60 PRINT TAB(J);“/    \”
70 FOR K = 1 TO 1000:NEXT K:CLS
80 PRINT TAB(J+1);“–O–”
90 PRINT TAB(J+1);“<  >”
100 PRINT TAB(J+1);“/    \”
110 FOR K = 1 TO 1000:NEXT K:NEXT J
```

2.

```
10 REM *GUESSING GAME*
20 INPUT “TYPE ANY NUMBER”;M
30 IF M = 7 THEN PRINT “YOU ARE LUCKY” : GOTO 50
40 PRINT “PLEASE TRY AGAIN” : GOTO 20
50 PRINT “THANK UOY” : END
```

3.

```
05 CLS
10 INPUT “A 5 LETTER WORD”;J$
20 FOR A = 1 TO 32
30 PRINT TAB(A);J$;TAB(70-A);J$
40 NEXT A
50 PRINT TAB(35);T$:GOTO 5
```

4.

```
10 REM **FACE MAKER*
```

```
20 CLS
30 PRINT "       •       "
40 PRINT "    •     •    "
50 PRINT "  •          • "
60 PRINT "  • • • • • •  "
70 PRINT "  •  Θ   Θ  •  "
80 PRINT "   •   v   •  "
90 PRINT "      = =    "
100 PRINT "    •      •"
110 END
```

5.

```
10 REM *EVEN NUMBERS TILL 120
20 FOR I = 2 TO 120 STEP 2
30 PRINT I:NEXT I
```

6.

```
10 REM *ODD NUMBERS TILL YOUR WISH*
20 INPUT "TYPE THE LAST LIMIT";N
30 FOR I = 1 TO N STEP 2
40 PRINT I:NEXT I
```

7.

```
10 REM *PRIME NUMBERS*
20 CLS
30 PRINT 2;
40 FOR I = 3 TO 250
50 FOR M = 2 TO I-1
60 IF I/M = INT(I/M) THEN 90
70 NEXT M
80 PRINT I;
90 NEXT I:END
```

8.

```
10 REM *PROGRAM TO CHECK
20 REM WHETHER A NUMBER
30 REM IS ODD OR EVEN*
40 INPUT N
50 LET M = N/2
60 IF M = INT(M) THEN PRINT "EVEN" ELSE PRINT "ODD"
70 END
```

GLOSSARY

Algorithm	A sequence of steps that specify the solution to a given problem.
Alphanumeric	Text which contains letters and numbers. It can include digits 0–9 and letters A–Z or any combination.
ALU	(Arithmetic Logic Unit) A part of central processing Unit which solves all arithmetical and logical calculations.
Assignment	The operation of giving a value to a variable.
BASIC	(Beginners All-purpose Symbolic Instruction Code) An easy to learn high level language.
Binary	A number system with a base of two. It consists of 2 digits — 0 or 1.
Bug	Program error. Bugs should be prevented rather than cured.
COBOL	(Common Business Oriented Language). A high-level language developed mainly for business applications.
Code	A set of rules outlining the way in which data may be represented within a computer system.
Command	A reserved word used to program a specific starting, executing a program etc.
Computer	An electronic equipment designed to automatically accept and store input data, process them and produce output results.
Constant	A value, written in a program, that does not change during the execution of the program.
CPU	(Central Processing Unit). It is the brain of any computer system.

Cursor	A little flashing dash on the monitor (CRT) that indicates where the next character will be displayed.
Data	Raw facts. All items of information a computer can process or generate — numbers, letters, symbols etc.
Debug	The process of eliminating errors (bugs) from a computer program.
Device	Any piece of computer equipment.
Expression	In an assignment statement, the value to the right of the equal sign.
Exponentiation	Raising a value to some power thus. 10 LET M = J ↑ 3 is the same as 10 LET M = J*J*J
Execute	To run a program by following the instructions given in it.
Escape	To interrupt execution of a program. Many computers have a control key labeled ESC.
File	An organized collection of related records.
Function key	A key that tells the computer to perform a specific action.
Fortran	(Formula Translation). A high-level mathematically oriented programming language used for scientific and engineering applications.
Flowchart	A pictorial representation that uses predefined symbols to describe the logic of a computer program.
GIGO	(Garbage In Garbage Out) Pertains to the facts that most computer errors are not machine errors, they are data erroes, caused by incorrect data input. Therefore incorrect input data leads to in accurate output.

Hardware	The various physical components of a computer system.
High-level-language	A computer language designed to facilitate giving instructions to the computer.
Interface	Electronic circuits that allow the connection of a specific device to the computer.
Information	The output given by a computer program.
Input	To transfer data from the keyboard or a mass storage device into the computers internal memory.
Integer	A whole number.
Joystick	An input device for entering. X-Y co-ordinates. It is commonly used with computer games.
Language	A means of communicating.
Logical operators	Symbols used to show a logical relationship between 2 data items.
Loop	A sequence of instructions that is executed repeatedly until a terminal condition occurs.
Linenumber	A number that defines each line of a computer program. The computer executes the program in linenumber order starting with the lowest number.
Memory	The internal hardware in the computer that stores information for further use.
Modem	A modulating and demodulating device that enables computers to communicate over telephone lines.
Monitor	A T.V. or cathode ray tube used to display computer information.
Numeric	Data consisting of only numbers.
Nested loops	Loop embedded within another loop.
Operator	A symbol representing any valid operation or values (ie. +(add), *(multiply) etc.)

Pascal	A high level language named after French scientist Blaise Pascal.
Program	A set of instructions telling a computer how to solve a given problem.
Programmer	A person who develops computer programs.
Programming	The process of developing a computer program.
Printer	A device for producing paper copies (hard copy) of data output by a computer.
Prompt	A symbol appearing on the screen that asks you to enter information.
Relational operators	The symbols , = and that may be used to indicate whether one value is larger, smaller, equal or not equal to another. hey are used in IF statements.
Record	An organized block of data.
RAM	(Random Access Memory).Internal memory of a computer which is the working memory and which has data, instruction, programs stored temporarily.
ROM	(Read only Memory). Internal memory of a computer which is a solid state storage chip which has certain programs fused into it by the manufacturer permanently.
Software	The programs & instructions used to control a computer.
Statement	A single instruction to a computer such as 10 LET L = 25
Syntax	A set of rules that defines the acceptable instructions of a computer language.
String	A sequence of characters.
Variable	A storage location in main memory whose contents may change during execution.

ANSWERS

EXERCISE – 1

I.

1. COMPUTER
2. CALCULATOR
3. INPUT
4. DATA
5. MEMORY
6. RECORD
7. OPERATORS
8. PRINTER
9. PROGRAM
10. DEBUGGING

II.

1. CHARLES BABBAGE
2. CONTROL UNIT
3. *
4. <>
5. INPUT
6. BUG
7. HARDWARE
8. PROGRAMMER

III.

1. Central Processing Unit
2. Arithmetic Logic Unit
3. Garbage In Garbage Out
4. Control Unit

IV.

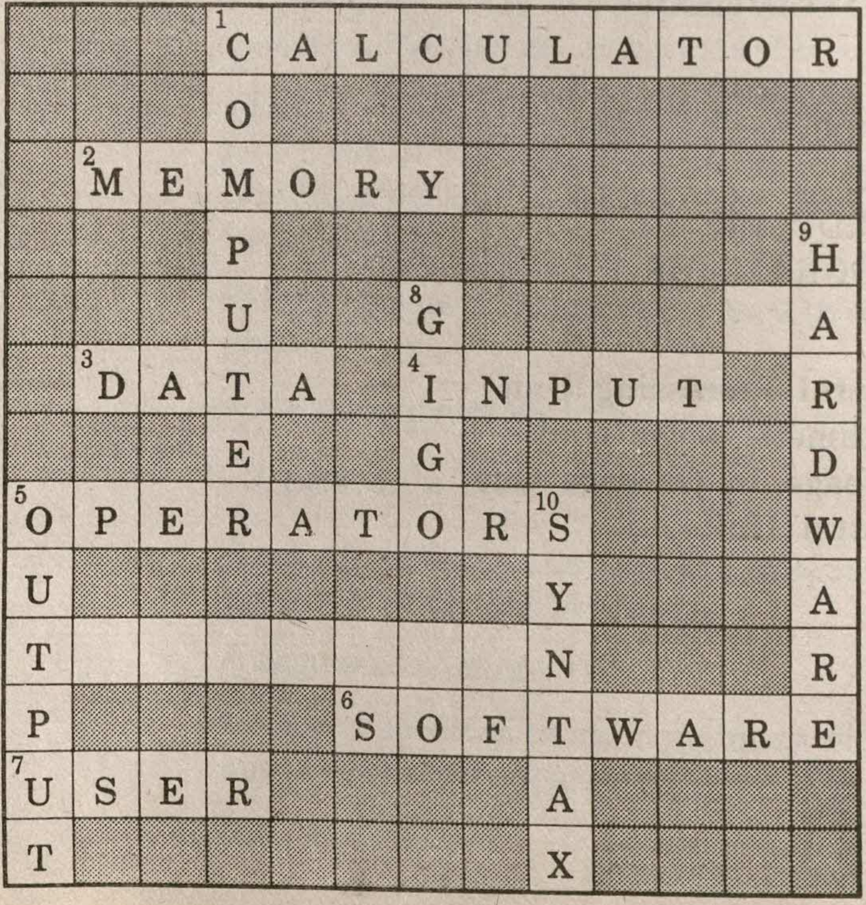
1. H
2. L
3. S
4. L
5. N
6. H

7. N
8. N
9. N
10. S

V.

1. A*B*C
2. (2*N)/M
3. Z*(Y ↑ 3)
4. (4*X)–(7*Y)
5. (A+B) 2+D
6. (X ↑ 2)+(Y ↑ 2)
7. (L+M)/(J+K)
8. (T–Q)*R+(P*T)
9. (X–Y)*(A+B)
10. ((E ↑ 2)+(F 3))/G

VI. Cross word answer



▨	▨	▨	1 C	A	L	C	U	L	A	T	O	R
▨	▨	▨	O	▨	▨	▨	▨	▨	▨	▨	▨	▨
▨	2 M	E	M	O	R	Y	▨	▨	▨	▨	▨	▨
▨	▨	▨	P	▨	▨	▨	▨	▨	▨	▨	▨	9 H
▨	▨	▨	U	▨	▨	8 G	▨	▨	▨	▨		A
▨	3 D	A	T	A	▨	4 I	N	P	U	T	▨	R
▨	▨	▨	E	▨	▨	G	▨	▨	▨	▨	▨	D
5 O	P	E	R	A	T	O	R	10 S	▨	▨	▨	W
U	▨	▨	▨	▨	▨	▨	▨	Y	▨	▨	▨	A
T								N	▨	▨	▨	R
P	▨	▨	▨	▨	6 S	O	F	T	W	A	R	E
7 U	S	E	R	▨	▨	▨	▨	A	▨	▨	▨	▨
T	▨	▨	▨	▨	▨	▨	▨	X	▨	▨	▨	▨

EXERCISE – 2

I.

1. False
2. False
3. True
4. True
5. False
6. False
7. False
8. True

II.

1. To get the line numbers automatically on the screen.
2. To get the answer of the program.
3. To get the program again on the screen.
4. To clear the screen.

III.

1. BASIC,COBOL
2. RUN,AUTO
3. M,B
4. LET,PRINT
5. 8,2

IV. Crossword answer.

▨	▨	▨	▨	▨	1 K	E	Y	B	O	7 A	R	D
▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	U	▨	▨
9 C	▨	▨	▨	2 C	H	A	R	A	C	T	E	8 R
O	▨	▨	▨		▨	▨	▨	▨	▨	O	▨	U
3 N	U	M	E	R	I	C	▨	▨	10 V	▨	▨	N
S	▨	▨	▨		▨	▨	11 S	▨	A	▨	▨	▨
T	▨	▨	4 M	O	N	I	T	O	R	▨	▨	▨
A	▨	▨	▨		▨	▨	R	▨	I	▨	▨	▨
N	▨	▨	▨	▨	▨	▨	I	▨	A	▨	▨	▨
T	▨	▨	▨	▨	▨	▨	N	▨	B	▨	▨	▨
S	▨	▨	▨	▨	▨	▨	G	▨	5 L	I	S	T
▨	▨	▨	▨	▨	▨	▨	▨	▨	E	▨	▨	▨
▨	▨	▨	▨	▨	▨	▨	6 B	A	S	I	C	▨

EXERCISE – 3

I.

Type and find it on the computer.

II.

1.
```
10 PRINT 7+3-2
20 END
```

2.
```
10 PRINT (5+6*9)*8
20 PRINT "DEBUG"
30 END
```

3.
```
10 PRINT 10*100*1000
20 PRINT "Rs. 5"
30 END
```

4.

```
100 PRINT “PRINT”
120 PRINT “PRINTS”
130 PRINT “PRINT”
```

5.

```
10 PRINT “STATEMENT”
20 PRINT “&”
30 PRINT “COMMAND”
40 END
```

6.

```
100 PRINT “GOOD”;“MORNING”
200 PRINT “SALAAM”
300 PRINT “NAMASKAR”
400 END
```

7.

```
10 CLS
20 PRINT “I LOVE“
30 PRINT
40 PRINT “INDIA”
50 END
```

8.

```
11 PRINT “*****”
12 PRINT “ *** ”
13 PRINT “  *  ”
14 END
```

III.

1.

```
30 END
```

2.

```
10 CLS
30 END
```

3.

```
10 PRINT “DELHI” _’_ “IS”
20 PRINT “THE CAPITAL”
30 PRINT “OF INDIA”
```

4.

```
10 PRINT 11 * 22 * 33 * 44
```

5.

```
100 PRINT 100 ↑ 2
200 END
```

IV.

1.

```
10 PRINT 700+70+7
20 END
```

2.

```
10 PRINT 30*3000
20 PRINT 40/4
30 END
```

3.

```
10 PRINT 950/100
20 END
```

4.

```
10 PRINT “X”,“X”,“X”,“X”
20 END
```

[Type your names instead of X]

5.

```
10 PRINT “S”,“A”
20 END
```

[Where S = names of your School A = its Address.]

6.

```
10 PRINT 2800*(15/100)
20 END
```

7.

```
10 PRINT “AREA =”;70 2
20 END
```

8.

```
10 PRINT 18.50 + 20 + 25
20 PRINT 100-(18.50+20+25)
30 END
```

9.

```
10 PRINT (10*53) + (92*0.80)
20 END
```

10.

```
10 PRINT 4*5.55
20 END
```

11.

```
10 PRINT 640-40
20 END
```

12.

```
10 PRINT “%%%%%%%%%%%%”
20 PRINT “%         %”
30 PRINT “%%%%%%%%%%%%”
40 END
```

13.

```
10 PRINT “ - - - - - - - - - - - - - - -”
20 PRINT “-                                 -”
30 PRINT “- - - - - - - - - - - - - - -”
40 PRINT “-                 O                 -”
50 PRINT “- - - - - - - - - - - - - - -”
60 PRINT “-                                 -”
70 PRINT “- - - - - - - - - - - - - - -”
80 PRINT “- - -”
90 PRINT “- - -”
100 PRINT “- - -”
110 PRINT “- - -”
120 PRINT “- - -”
130 PRINT “- - -”
140 PRINT “- - -”
150 PRINT “- - -”
160 PRINT “- - -”
170 PRINT “- - -”
180 PRINT “- - -”
190 PRINT “- - -”
200 END
```

EXERCISE - 4

I. Type & find it on the computer.

II.

1.

```
10 REM **DIVIDE TWO NUMBERS**
20 LET X = 64:LET Y = 7
30 PRINT X/Y
40 END
```

2.

```
10 REM *TO PRINT THE MESSAGE*
20 LET K$ = "WORRY GIVES SMALL THINGS"
30 LET J$ = "BIG SHADOWS"
40 PRINT K$,J$
50 END
```

3.

```
10 LET B = 100
20 LET E = 8 * B
30 PRINT E
40 END
```

4.

```
10 REM *TOTAL MARKS OF 4 SUBJECTS*
20 LET B = 80:LET F = 90:LET P = 60:C = 70
30 PRINT B+F+P+C
40 END
```

5.

```
10 LET  M$ = "MOTHER"
20 REM _ _ _ _ _
30 PRINT M$
40 PRINT "MAA"
50 END
```

6.

```
10 PRINT 10
20 LET G$ = "*'s"
30 PRINT G$,"IN THE"
40 PRINT "SKY"
50 END
60 REM *REMARK*
```

7.

```
10 REM *AREA OF A SQUARE*
20 LET SD = 70
30 PRINT SD*SD
40 END
```

8.

```
100 LET X$ = "ABC"
200 LET Y$ = "OF COMPUTER"
300 LET Z$ = "1989"
400 PRINT X$,Y$
500 PRINT Z$
600 END
```

9.

```
10 LET L$ = "LADY"
20 LET F$ = "FINGER"
30 PRINT L$;F$
40 END
```

10.

```
100 LET R = 6
200 LET S = 7
300 LET T = 8
400 PRINT R*S*T
500 END
```

III.

1.

```
30 PRINT M$,M$
```

2.

```
10 CLS
20 LET N$ = "BA"
30 PRINT N$
40 PRINT N$
50 PRINT N$
```

[where BA can be any name]

3.

```
10 LET X = 40:LET Y = 30
20 LET Z = X*Y:PRINT Z:END
```

4.

```
10 LET CP = 450
30 LET PR = SP - CP:PRINT PR
```

5.

```
10 LET A = 99:LET B = 79 : LET C = 89
30 PRINT D/3
```

IV.

1. PRINT	an executable ststement
2. PRINTER	an output device
3. REM	a non executable statement
4. NEW	a command
5. KEYBOARD	an input device
6. ↑	a mathematical operator
7. PROGRAMMER	a part for liveware
8. BASIC	a programming language.
9. MM$	a string variable
10. U	a character
11. CONTROL UNIT	a part of CPU
12. K	a unit for measuring the power of the computer.
13. Equal to sign (=)	a relational operator
14. "TIT FOT TAT"	a string constant
15. 20	a linenumber

V.

1.

```
10 REM *AREA OF A SQUARE*
20 LET S = 4
30 LET A = S*S
40 PRINT A
50 END
```

2.

```
100 REM *AREA OF A RECTANGLE*
200 LET L = 10
300 LET B = 20
400 LET A = L*B
500 PRINT A
600 END
```

3.

```
10 LET A = 4
20 PRINT A*2
30 PRINT A*3
```

```
40 PRINT A*4
50 END
```

4.

```
 4 LET C = 6
 8 LET M = C*2
12 PRINT "MOTTU'S AGE=";M
16 END
```

5.

```
10 LET R = 25
20 LET S = 35
30 LET T = 45
40 LET SUM = R+S+T
50 LET PRO = R*S*T
60 PRINT SUM,PRO
70 END
```

6.

```
100 REM *PERIMETER OF A SQUARE*
200 LET P = 200
300 LET E = P*4
400 PRINT "PERIMETER=";E
500 END
```

7.

```
10 LET E = 90
20 LET M = 100
30 LET H = 35
40 LET TOT = E+M+H
50 LET AV = TOT/3
60 PRINT "TOTAL",TOT
70 PRINT "AVERAGE",AV
80 END
```

8.

```
100 LET S = 60
200 LET T = 8
300 LET D = S*T
400 PRINT D
500 END
```

9.

```
10 LET A = 2000
20 LET D = 20/100
```

```
30 PRINT A+(A*D)
40 END
```

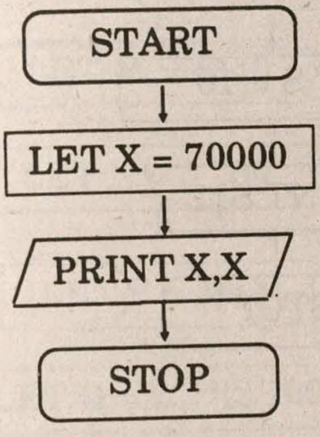
10.
```
10 E = 60
20 P = 40
30 LET O = E/P
40 LET D = O*12
50 PRINT D
60 END
```

VI. Cross word answer

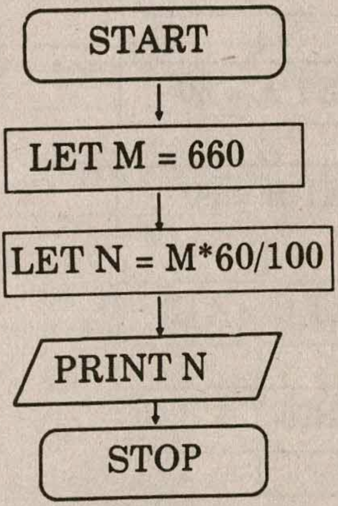
▨	1 D	E	6 B	U	G	▨	▨	7 C	▨	▨
▨	R	▨	U	▨	▨	2 C	O	L	O	N
▨	Y	▨	G	▨	8 L	▨	▨	S	▨	▨
▨	R	▨	▨	▨	E	▨	▨	▨	▨	▨
3 O	U	T	9 P	U	T	▨	▨	▨	▨	10 R
▨	N	▨	R	▨	▨	▨	▨	▨	▨	E
▨	S	▨	O	▨	▨	▨	▨	▨	▨	M
▨	▨	▨	G	▨	11 E	▨	▨	▨	▨	▨
▨	▨	4 P	R	I	N	T	▨	▨	▨	▨
▨	▨	▨	A	▨	D	▨	▨	▨	▨	▨
▨	5 I	B	M	▨	▨	▨	▨	▨	▨	▨

EXERCISE – 5

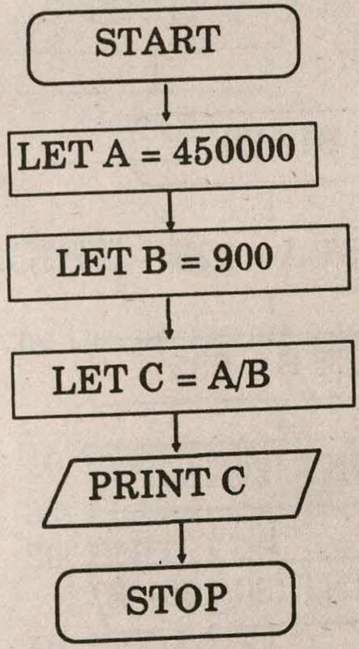
I. 1.



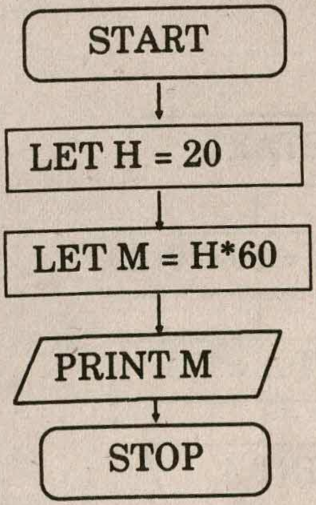
I. 2.



I. 3.



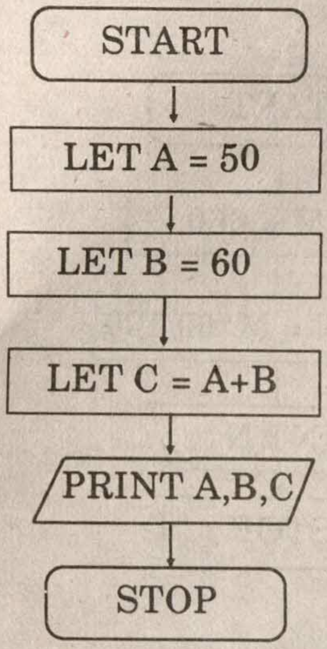
I. 4.



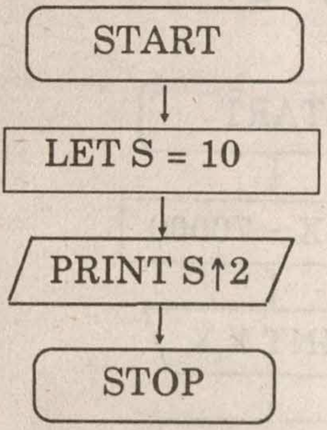
II.

1. In a flowchart, the flow of data should always be downwards.
2. Every flowchart has a terminal symbol in the beginning and at the end.
3. There is only one entry into a process box.
4. There are two exits for every decision box.
5. The first formal flowchart was drawn by John Von Newman.

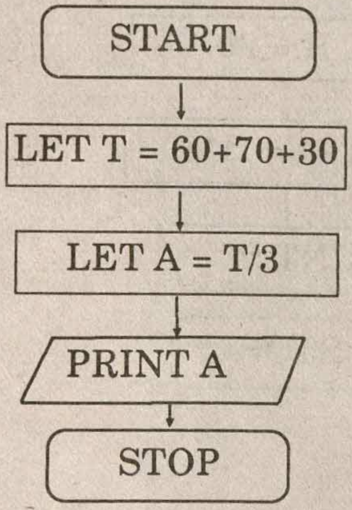
III. 1.



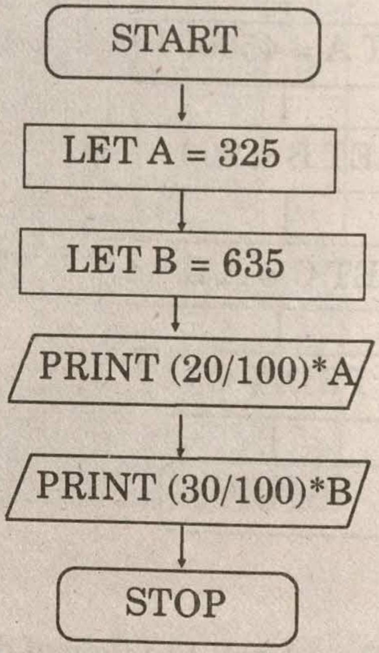
III. 2.



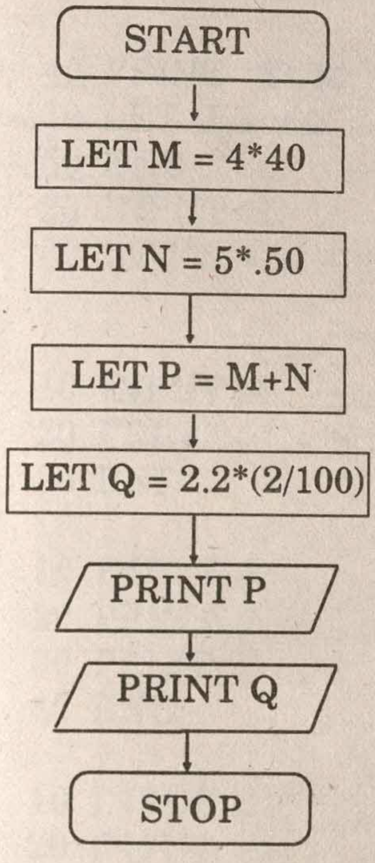
IV. 1.



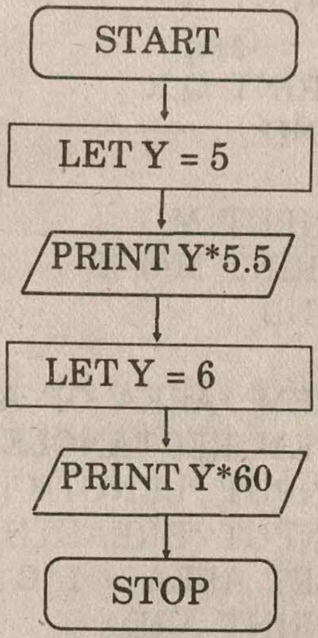
IV. 2.



IV. 3.



IV. 4.



EXERCISE - 6

I. Type the programs and get there output on the computer.

II.

1.
```
10 INPUT P
20 PRINT P+100
30 END
```

2.
```
10 INPUT R
20 PRINT R*6
30 END
```

3.
```
10 INPUT A,B
20 PRINT A+B
30 END
```

4.

```
10 INPUT Q
20 R = Q+Q
30 PRINT Q,R
40 END
```

5.

```
10 INPUT M
20 PRINT M/6,M
30 END
```

6.

```
10 REM *AREA OF A
20 REM RECTANGLE*
30 INPUT “LENGTH”;L
40 INPUT “BREADTH”;B
50 LET AREA = L*B
60 PRINT AREA
70 END
```

III.

1.

```
20 PRINT X,Y
10 INPUT X,Y
30 END
```

2.

```
40 END
30 PRINT Z
20 LET Z = L*L
10 INPUT L
```

3.

```
20 PRINT “LOVE COMPUTERS”
10 PRINT “DO YOU”
50 END
30 INPUT “TYPE YES OR NO”;N$
40 PRINT “O.K.THANK YOU!”
```

4.

```
10 REM ENQUIRY
60 END
40 PRINT “YOUR ARE LUCKY TO BE IN”;N$
30 INPUT N$
20 PRINT “YOUR SCHOOL’S NAME PLEASE”
```

```
50 PRINT “OH! WHAT A GOOD SCHOOL”
```

5.

```
40 PRINT “SUM OF IT=”;S
10 LET J = 60
50 END
20 INPUT K
30 LET S = J + K
```

IV.

1.

```
10 INPUT K
20 PRINT K + K
30 END
```

2.

```
10 INPUT P
20 PRINT P ↑ 2
30 PRINT P ↑ 3
40 END
```

3.

```
10 INPUT PR
20 INPUT QU
30 PRINT “TOTAL COST”,PR * QU
40 END
```

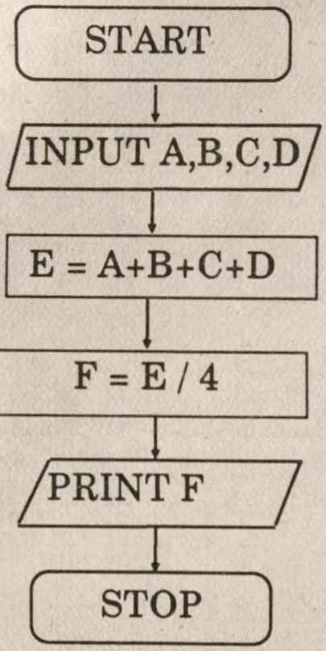
4.

```
10 INPUT “NAME”;N$
20 INPUT “AGE”;A
30 INPUT “NATIONALITY”;Y$
40 INPUT “ADDRESS”;AD$
50 PRINT N$
60 PRINT A
70 PRINT Y$
80 PRINT AD$
90 END
```

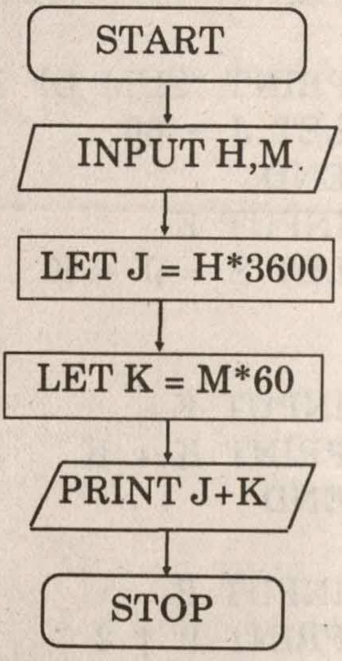
5.

```
10 INPUT J,K
20 LET P = J*K
30 LET M = 10*P
40 PRINT J,K,M
50 END
```

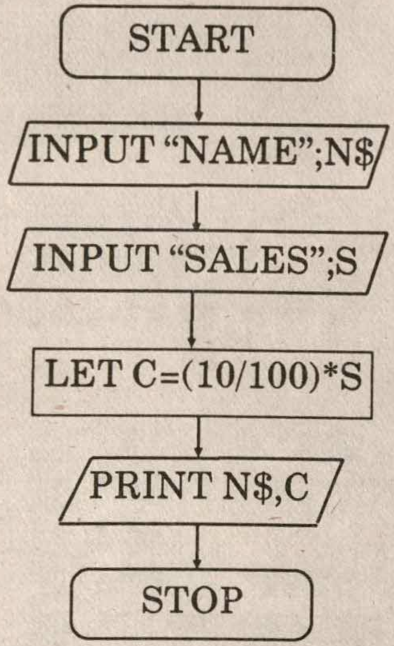
V. 1.


START
INPUT A,B,C,D
E = A+B+C+D
F = E / 4
PRINT F
STOP

V. 2.


START
INPUT H,M
LET J = H*3600
LET K = M*60
PRINT J+K
STOP

V. 3.


START
INPUT “NAME”;N$
INPUT “SALES”;S
LET C=(10/100)*S
PRINT N$,C
STOP

EXERCISE – 7

I. Type the programs and get the output on the computer.

II.

1.
```
10 REM *REM*
20 PRINT 6+6
30 GOTO 20
```

2.
```
10 REM *JUMP*
20 LET V = 2
30 PRINT “V”, V
40 GOTO 10
```

3.
```
10 INPUT X,Y
20 PRINT (X/Y) + X
30 GOTO 10
```

4.
```
10 LET P = 3000
20 LET D = 20
30 LET K = P + D
40 PRINT K,P
50 GOTO 40
```

5.
```
10 INPUT M
20 PRINT “M =”,M
30 GOTO 10
40 END
```

III

1.
```
10 PRINT “MTL”
20 GOTO 10
```

2.
```
10 CLS
20 PRINT “THIS WILL NEVER STOP”
30 GOTO 10
```

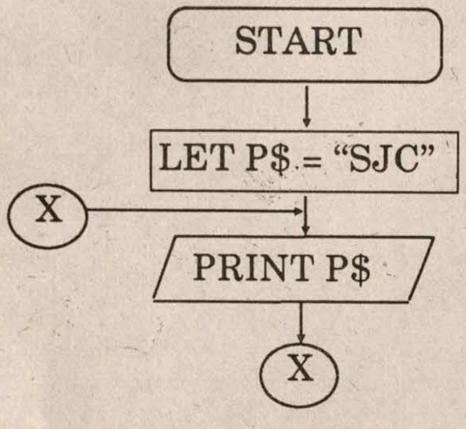
3.
```
10 PRINT “*”;
```

```
20 GOTO 10
```

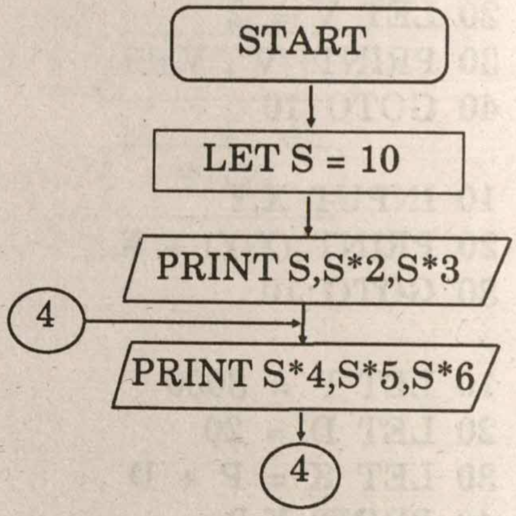
4.

```
10 INPUT "NAME OF CUSTPMER"; N$
20 INPUT "LIST PRICE"; L
30 LET D = L * (20/100) : PRINT N$
40 PRINT "AMOUNT OF DISCOUNT"; D
50 PRINT "SELLING PRICE",L–D
60 END
```

IV. 1.



IV. 2.



EXERCISE - 8

I. Types these programs & get the output on the computer.

II.

1.

```
10 LET C = 8.65
20 PRINT SQR(C), INT(C)
30 END
```

2.

```
10 LET S$ = "SJC"
20 PRINT TAB(18,12); S$
30 END
```

[For BBC micro computer]

2.

```
10 LET S$ = "SJC"
20 LOCATE 12,39
30 PRINT S$
```

```
40 END
```

[For IBM PC]

3.

```
10 INPUT G
20 PRINT SGN(G)
30 END
```

4.

```
10 PRINT TAB(4);“*”
20 PRINT TAB(3);“***”
30 PRINT TAB(2);“-----”
40 END
```

5.

```
10 LET G$ = “GEM”
20 INPUT “ROW,COLUMN”;R,C
30 PRINT TAB(C,R);G$
40 END
```

[For BBC micro computer]

5.

```
10 LET G$ = “GEM”
20 INPUT “ROW,COLMN”;R,C
30 LOCATE R,C
40 PRINT G$
50 END
```

[For IBM PC]

6.

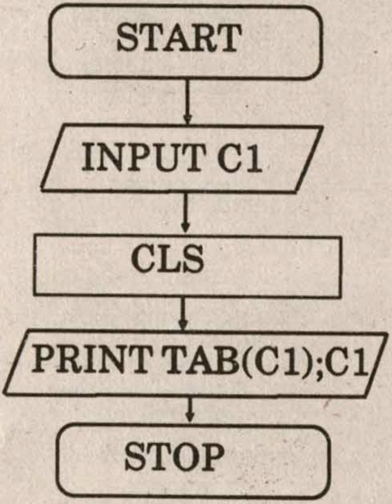
```
10 LET D = 98
20 LET E =  89
30 PRINT TAB(6);D;TAB(16);E
40 END
```

7.

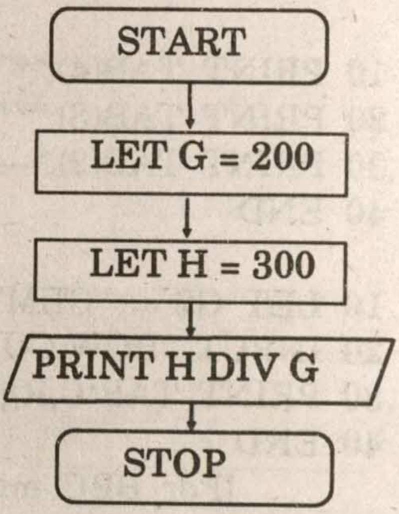
```
10 CLS
20 PRINT “BBBBB”
30 PRINT TAB(2); “AAAA”
40 PRINT TAB(3); “SSS”
50 PRINT TAB(4); “II”
60 PRINT TAB(5); “C”
70 END
```

8.
```
10 INPUT X
20 Y = INT(X)
30 PRINT SQR(Y)
40 END
```

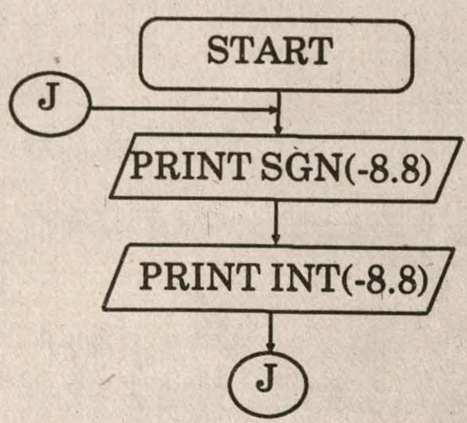
III. 1.



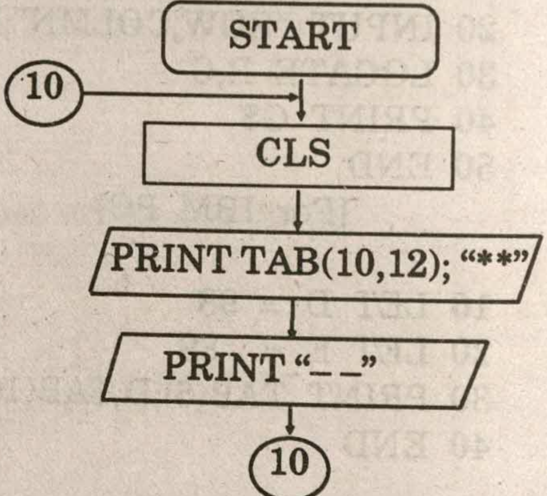
III. 2.



III. 3.



III. 4.



EXERCISE – 9

I. Type & get the answer.

II.

1.
```
40 PRINT LEN(A$)
50 END
```

2.

```
30 LET D$ = “WRITER”
40 PRINT C$ + D$
```

3.

```
30 LET M$ = LEFT$(J$,4)
40 PRINT M$
```

4.

```
30 PRINT LEFT$ (Z$,1)
40 PRINT RIGHT$(Z$,1)
```

III.

1.

```
10 INPUT L$
20 PRINT LEN(L$)
30 END
```

2.

```
10 G$ =“LUCKNOW”
20 PRINT LEFT$(G$,4)
30 PRINT RIGHT$(G$,4)
40 END
```

3.

```
10 INPUT “ANYWORD”; W$
20 PRINT “FIRST CHARACTER”, LEFT$(W$,1)
30 PRINT “LAST CHARACTER”, RIGHT$(W$,1)33
40 END
```

4.

```
10 INPUT “FIRST NAME”;F$
20 INPUT “SIR NAME”;S$
30 LET E = LEN(F$)
40 LET F = LEN(S$)
50 PRINT E + F
60 END
```

5.

```
10 LET T$ = “TIME”
20 PRINT LEFT$(T$,1)
30 PRINT LEFT$(T$,2)
40 PRINT LEFT$(T$,3)
50 PRINT RIGHT$(T$,1)
60 PRINT RIGHT$(T$,1)
70 PRINT RIGHT$(T$,3)
```

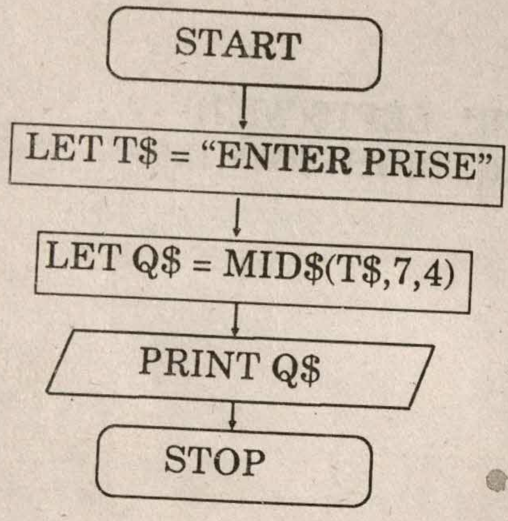
```
80 PRINT MID$(T$,2,1)
90 PRINT MID$(T$,3,1)
100 PRINT MID$(T$,2,2)
110 PRINT MID$(T$,1,4)
120 END
```

6.
```
10 LET D$ = "LADYFINGER"
20 PRINT LEFT$(D$,4)
30 PRINT RIGHT$(D$,6)
40 PRINT MID$(D$,6,2)
50 END
```

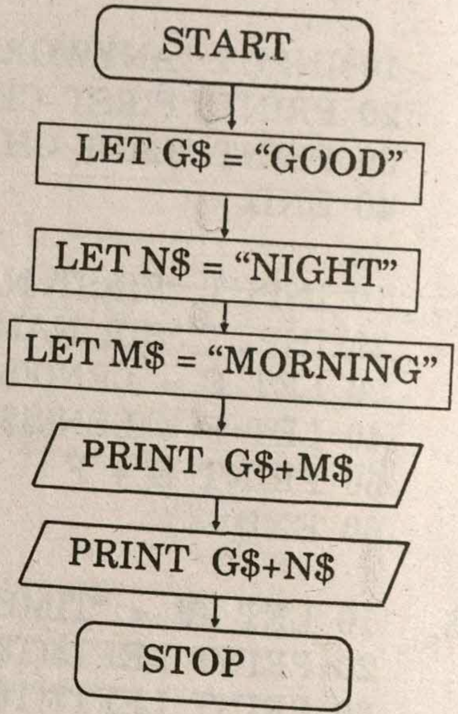
7.
```
10 INPUT "TYPE DATE AS MM/DD/YY";A$
20 LET B$ = MID$(A$,4,2)
30 PRINT B$
40 END
```

IV

1.



2.



EXERCISE - 11

I. Type and find the output on the computer.

II. Debugging

1.
```
10 REM TO CHECK WHETHER 2 NUMBER'S ARE =
20 LET O = 10 : LET P = 20
30 IF O = P THEN PRINT "YES" : GOTO 50
40 PRINT "NO"
50 END
```

2.
```
10 INPUT Q$
20 IF LEN(Q$) >= 10 THEN PRINT  "TOO LONG WORD"
30 END
```

3.
```
20 LET T = 5
100 PRINT "*************************", "GO"
200 T = T + 1
300 IF T = 10 THEN END ELSE GOTO 100
```

4.
```
10 REM NATURAL NUMBERS 100 - 200
20 KD = 98
30 KD = KD + 2
40 PRINT KD
50 IF KD < 200 THEN 30
60 END
```

III.

1. False
2. False
3. False
4. False
5. False
6. True
7. False
8. True
9. False
10. True

IV.

1.

```
20 REM OF 2 STRINGS R = OR UNEQUAL
30 INPUT T$ : INPUT U$
40 IF LEN(T$) = LEN(U$) THEN 60
```

2.

```
20 LET LL = 11
30 LET LL = LL – 1
40 PRINT LL
50 IF LL = 0 THEN END
60 GOTO 30
```

3.

```
20 W = 0
30 W = W + 1
40 PRINT "XYZ"
50 IF W <> 100 THEN 30
```

(Type your name in place of XYZ)

4.

```
30 CLS
40 INPUT "TYPE ANY WORD"; Q$
50 LET QM = LEN(Q$)
60 IF QM > 5 THEN PRINT Q$
70 GOTO 30
```

5.

```
20 INPUT "AGE";A
30 IF A >= 18 THEN PRINT "MAJOR" ELSE PRINT "MINOR"
```

V. Programs:–

1.

```
10 INPUT X
20 IF X > 100 THEN PRINT X
30 GOTO 10
```

2.

```
10 INPUT "TYPE YOUR NAME"; N$
20 INPUT "TYPE YOUR AGE"; M
30 IF M >= 18 THEN PRINT N$; "IS A MAJOR"
40 GOTO 10
50 END
```

3.

```
10 C = –1
20 C = C + 2
```

```
30 PRINT C
40 IF C = 29 THEN 60
50 GOTO 30
60 END
```

4.

```
 5 CLS
10 INPUT “NAME”; E$
20 INPUT “MARKS IN 4 SUBJECTS”; A,B,C,D
30 TOT = A+B+C+D
40 IF TOT >= 250 THEN PRINT E$; “HAS PASSED”: GOTO 60
50 PRINT E$; “HAS FAILED”
60 GOTO 5
```

5.

```
10 LET B = 110
20 LET B = B–10
30 PRINT B;
40 IF B = 10 THEN END
50 GOTO 20
```

6.

```
10 INPUT “TYPE ANY 2 NUMBERS”; X,Y
20 SUM = X+Y
30 IF SUM > 50 THEN PRINT X*Y: GOTO 50
40 PRINT X–Y
50 END
```

7.

```
10 CLS
20 INPUT “TYPE ANY WORD”; W$
30 INPUT “HOW MANY TIMES TO BE PRINTED”; T
40 T = 0
50 T = T + 1
60 PRINT W$
70 IF T = F THEN END
80 GOTO 50
```

8.

```
10 INPUT “A NUMBER”; R
20 IF R > 0 THEN PRINT “POSITIVE” : END
30 IF R < 0 THEN PRINT “NEGATIVE” : END
40 PRINT “ZERO” : END
```

9.

```
10 CLS
20 INPUT “TYPE ANY NUMBER”; N
30 LET M = N/2
40 IF M = INT(M) THEN PRINT “THE NUMBER IS EVEN”
                ELSE PRINT “THE NUMBER IS ODD”
50 END
```

10.

```
10 INPUT “GUESS ANY NUMBER BETWEEN 1 AND 50”; X
20 IF X = 39 THEN PRINT “LUCKY!” : END
30 PRINT “SORRY! TRY AGAIN”
40 GOTO 10
```

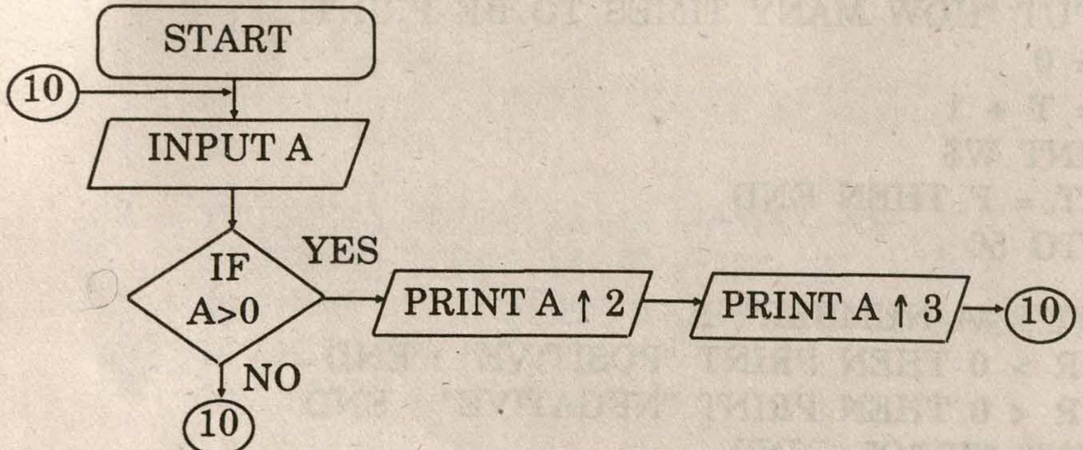
11.

```
10 INPUT “NAME”; A$
20 INPUT “AGE”; G
30 INPUT “NATIONALITY”; NA$
40 IF G >= 18 AND NA$ = “INDIAN” THEN PRINT
          “YOU CAN VOTE” : GOTO 10
50 PRINT “SORRY, YOU CANNOT VOTE” : GOTO 10
```

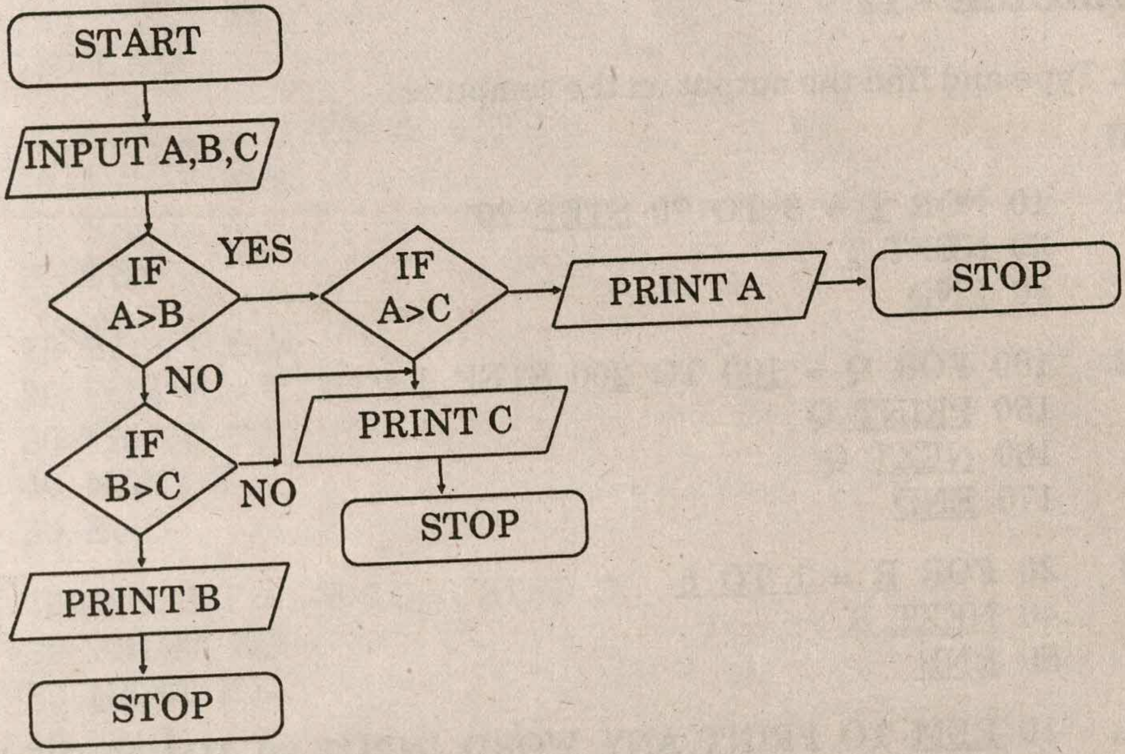
12.

```
10 INPUT “GIVE THE MEASUREMENT OF ANGLE" ; AN
20 IF AN > 90 THEN PRINT " OBTUSE ANGLE"
30 IF AN < 90 THEN PRINT " ACUTE ANGLE"
40 IF AN = 90 THEN PRINT " RIGHT ANGLE”
50 END
```

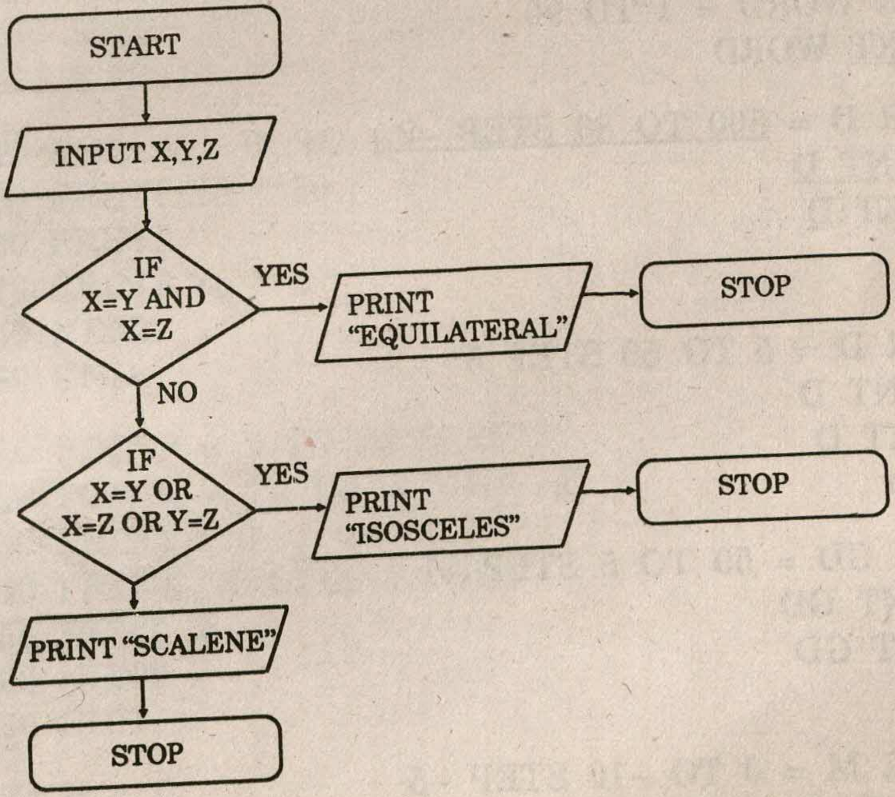
VI. 1.



2.


START
INPUT A,B,C
IF A>B
YES
IF A>C
PRINT A
STOP
NO
PRINT C
IF B>C
NO
STOP
PRINT B
STOP

3.


START
INPUT X,Y,Z
IF X=Y AND X=Z
YES
PRINT "EQUILATERAL"
STOP
NO
IF X=Y OR X=Z OR Y=Z
YES
PRINT "ISOSCELES"
STOP
PRINT "SCALENE"
STOP

EXERCISE - 12

I. Type and find the output on the computer.

II.

1.

```
10 FOR T = 6 TO 70 STEP 20
30 NEXT T
40 END
```

2.

```
100 FOR Q = 100 TO 200 STEP 1
150 PRINT Q
160 NEXT Q
170 END
```

3.

```
20 FOR R = 1 TO 5
40 NEXT R
50 END
```

4.

```
10 REM TO PRINT ANY WORD INPUT 90 TIMES ON THE
   SCREEN
20 INPUT T$
30 FOR WORD = 1 TO 90
50 NEXT WORD
```

5.

```
30 FOR D = 800 TO 80 STEP -2
40 PRINT D
50 NEXT D
```

III.

1.

```
10 FOR D = 5 TO 50 STEP 5
20 PRINT D
30 NEXT D
40 END
```

2.

```
10 FOR GD = 50 TO 5 STEP -1
20 PRINT GD
30 NEXT GD
40 END
```

3.

```
100 FOR M = 0 TO -10 STEP -5
110 PRINT M + M
```

```
120 NEXT M
130 END
```

4.

```
  5 FOR N = 2 TO 22 STEP 2
 10 PRINT N,N
 15 NEXT N
 20 END
```

5.

```
10 INPUT L,M
20 FOR J = L TO M
30 PRINT “**”
40 NEXT J
50 END
```

6.

```
10 FOR KT = –6 TO 6 STEP 2
20 PRINT KT
30 NEXT KT
40 END
```

7.

```
100 FOR H = 3 TO 33 STEP 3
120 PRINT H
130 NEXT H
140 END
```

8.

```
10 FOR Z5 = 5 TO 100 STEP 10
20 PRINT Z5 * 30
30 PRINT
40 PRINT Z5 * 2
50 NEXT
60 END
```

9.

```
10 FOR P = 2 TO 20 STEP 2
20 FOR Q = 20 TO 2 STEP –2
30 PRINT P,Q
40 PRINT “NESTED LOOPS”
50 NEXT Q
60 NEXT P
70 END
```

10.

```
10 K = 0
20 REM SUM OF THE CUBES OF FIRST FIVE
```

```
25 REM NATURAL NUMBERS
30 FOR I = 1 TO 5
40 P = I ↑ 3
50 K = K + P
60 NEXT I
70 PRINT K
80 END
```

IV

1.
```
10 FOR L = 1 TO 1000
20 PRINT L
30 NEXT L
40 END
```

2.
```
10 FOR X = 2 TO 20 STEP 2
20 PRINT X
30 NEXT X
40 END
```

3.
```
10 FOR J = 1 TO 39 STEP 2
20 PRINT J
30 NEXT J
40 END
```

4.
```
10 INPUT "TYPE YOUR NAME"; N$
20 INPUT "TYPE YOUR AGE"; A
30 FOR I = 1 TO 100
40 PRINT N$
50 PRINT A
60 NEXT I
END
```

5.
```
10 INPUT "TYPE ANY WORD"; D$
20 INPUT "HOW MANY TIMES TO BE PRINTED"; T
30 FOR I = 1 TO T
40 PRINT D$
50 NEXT I
END
```

6.

```
10 FOR I = 1 TO 20
20 PRINT 5 * I
30 NEXT I
40 END
```

7.

```
10 INPUT R
20 FOR V = 1 TO 10
30 PRINT R * V
40 NEXT V
50 END
```

8.

```
 5 SUM = 0
10 FOR T = 1 TO 10
20 M = 3 * T
30 SQ = M ↑ 2
40 SUM = SUM + SQ
50 NEXT T
60 PRINT SUM
70 END
```

9.

```
10 A = 0
20 FOR I = 1 TO 5
30 INPUT NUM
40 A = A + NUM
50 NEXT I
60 PRINT A/5
70 END
```

10.

```
10 FOR W = 1 TO 100
20 SUM = SUM + W
30 NEXT W
40 PRINT SUM
50 END
```

11.

```
10 TT = 0
20 FOR T = 1 TO 50
30 IF T > 20 THEN TT = TT + 1
40 NEXT T
50 PRINT TT
60 END
```

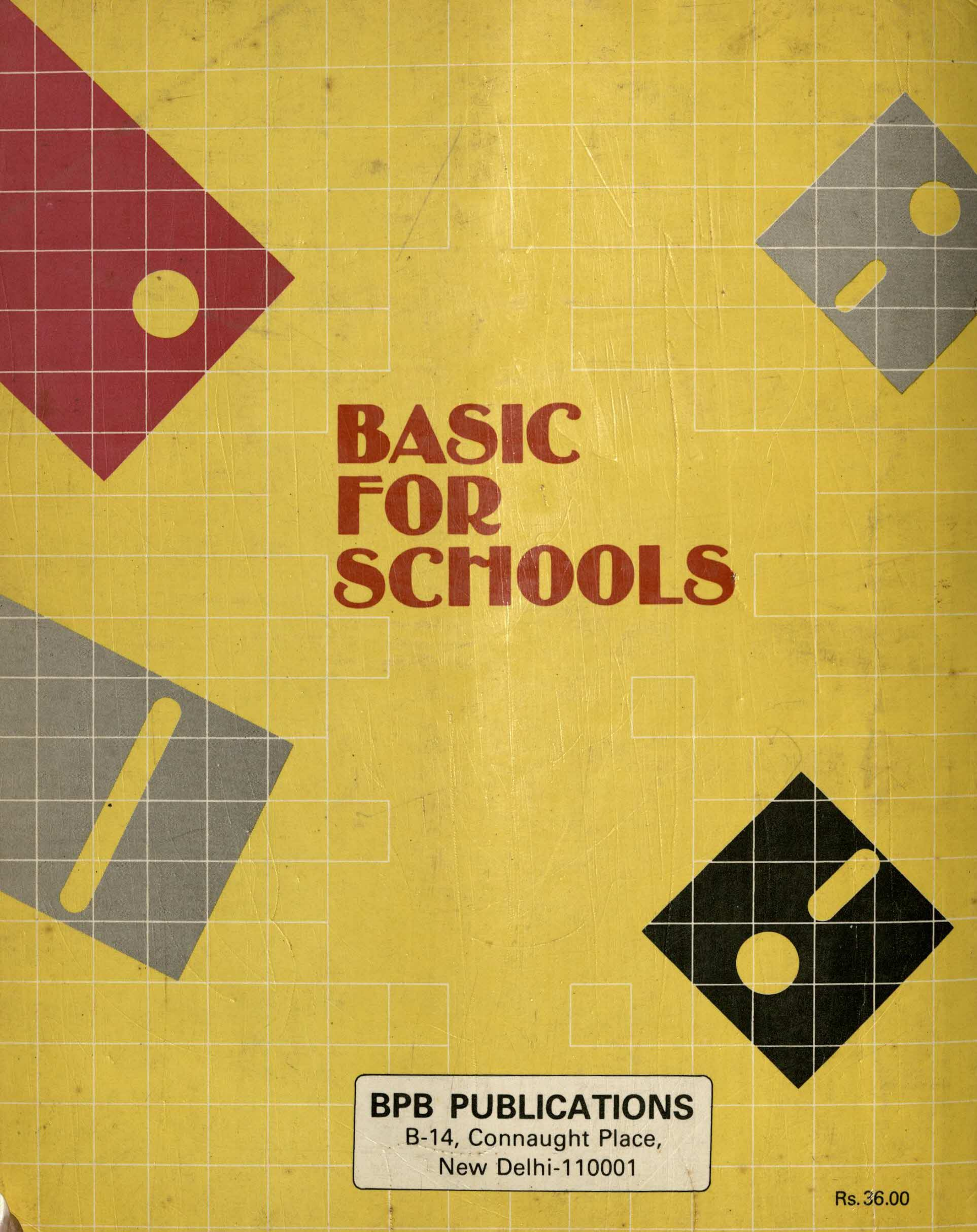
12.
```
10 FOR I = 5 TO 1 STEP –1
20 PRINT I * 8
30 NEXT I
40 END
```

13.
```
10 FOR S = 1 TO 6
20 FOR B = 1 TO 6
30 PRINT S * B
40 NEXT B
50 NEXT S
60 END
```

14.
```
10 FOR BP = 1 TO 4
20 PRINT SQR(BP)
30 PRINT BP ↑ 2
40 PRINT BP ↑ 3
50 NEXT BP
60 END
```

15.
```
 5 LET C = 0
10 FOR X = 1 TO 10
20 INPUT PR, N
30 COST = PR * N : C = C + COST
40 NEXT X
50 PRINT C
60 END
```

16.
```
10 INPUT N$
20 L = LEN(N$)
30 FOR D = L TO 1 STEP –1
40 PRINT MID$(N$,D,1);
50 NEXT D
60 END
```



BASIC FOR SCHOOLS

BPB PUBLICATIONS
B-14, Connaught Place,
New Delhi-110001

Rs. 36.00